Performance Prediction of an Image Management and Communication System for Cardiac Ultrasound

by

Dhananjay Rammohan Ragade

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1995

Author ....
Electrical Engineering and Computer Science
June 30, 1995

Certified by ....
Dr. David L. Tennenhouse
Associate Professor, Electrical Engineering and Computer Science
Thesis Supervisor

Certified by ....
Barry F. Hunt
Engineer, Hewlett-Packard Medical Products Group
Thesis Supervisor

Accepted by ....
Dr. Frederic R. Morgenthaler
Chairman, Commitee on Graduate Students

Performance Prediction of an Image Management and Communication System for Cardiac Ultrasound

by

Dhananjay Rammohan Ragade

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Engineering.

Abstract

A prediction of the performance of an Image Management and Communication System (IMACS) for a hospital's echocardiology lab is necessary prior to the construction of such a system. A discrete event simulation model of a hypothetical echolab IMACS is developed and used to draw conclusions about the expected network performance. An analysis of the storage side of the potential workload is also presented along with findings on how the network medium (switched 100VG-AnyLan or shared Fast Ethernet), bus speed of the storage transfer (normal SCSI-2 or fast-wide SCSI-2), and storage block size (4KB or 512 KB) affect the performance of the image management system.

Thesis Supervisor: Dr. David L. Tennenhouse
Title: Associate Professor, Electrical Engineering and Computer Science

Thesis Supervisor: Barry F. Hunt
Title: Engineer, Hewlett-Packard Medical Products Group

Acknowledgements

I would like to thank:

- Dr. David Tennenhouse for his guidance with the thesis, and especially his help in debugging the various models.
- Barry Hunt, of HP, for proving that the paperless office can work by editing the chapters of my thesis online in maker interchange format, and emailing the revisions to me.
- Dr. Steve Finn for introducing me to OPNET and allowing me to use it on the Sparcs. This project would have taken way more than ten times as long to complete if, God forbid, I had to work on the DECstations.
- Damian Danowski, of HP, for his helping me to understand various network technologies.
- Frame Technology Corp. for making what I believe is the most complete, most incredible document preparation system on the face of the planet: FrameMaker 4.
- Sun Microsystems for putting a real chip and a CD player in the Sparc5s.
- Most of all, I would like to thank my family for their love and support, especially my father Dr. Rammohan Ragade for sending me information whenever he found something interesting, and for encouraging me all the way; and my Mom for continually logging on and writing me email to see how I was doing.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Currently, physicians in the echo lab of a hospital's cardiology unit must view stored images on videotape in order to make a diagnosis based on ultrasound imaging. Usually, technicians perform the ultrasound exam and record the results on videotape for later review by the cardiologist. Watching the exam on video instead of attending the imaging session allows the cardiologist to more effectively use his or her time, and only see and hear the important saved sequences from the exam.

Although videotape allows cardiologist to diagnose at their convenience, reviewing ultrasound exams on videotape has a lot of problems. Compared to the original digital image, e.g. on an HP SONOS Ultrasound System, video tape has poor image quality. Furthermore, searching for particular segments is impracticable and adding annotations directly to the images is impossible. Moreover, a substantial amount of physician time is spent in reviewing merely because of the sequential analog format of the images—the physician has to fast forward or rewind many times to be able to find the specific frames of interest. Even with digital search software, searching for an individual patient can be tedious. [Feigenbaum(3), 378]

Cardiologists also have significant difficulty in trying to compare two or more studies recorded on video tape. Finding the correct study on a 2-hour VHS tape is time-consuming and the individual views must be locked in the cardiologists memory so that they can be compared with similar views in a subsequent study. Because of this inconvenience, comparing even two echocardiograms is rarely done. [Feigenbaum(3), 378] The odds of comparing multiple echocardiograms are even more remote.

1.1 The IMACS Solution

The Integrated Information Management (IIM) group at Hewlett-Packard Medical Products Group in Andover, Massachusetts is currently investigating the design opportunities for connecting the SONOS cardiac ultrasound machines together with workstations and storage devices on a network. Hewlett-Packard is considering the design of a net-

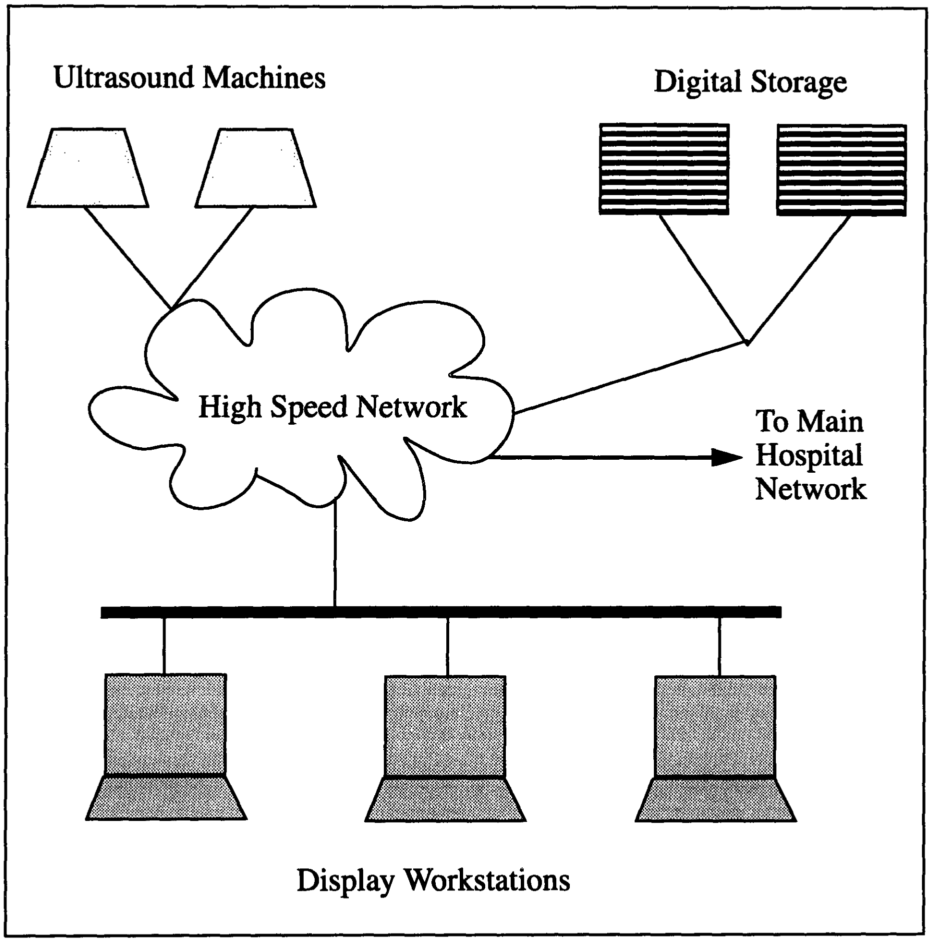

Figure 1.1: A Typical Image Management and Communication System MACS (IMACS) Network.

worked Image Management and Communication System (IMACS)[1] for a hospital's echo lab. Such a network would allow cardiologists to quickly view digitally stored ultrasound images on their workstations, process and annotate images, or view an imaging session in

1. The term IMACS is replacing the older term PACS, which mainly refers to systems designed for single images. IMACS are designed to handle sequences of images for animation.

real time from a remote location using a high powered graphics workstation. Figure 1.1 depicts the echo lab IMACS consisting of the HP SONOS cardiac ultrasound machines that generate the digital image data, the digital storage devices used to contain the image data, the graphic workstations used to display the images, and the high speed network medium used to connect all the devices in the echo lab.

The digital storage, processing, and retrieval of ultrasound images offers many benefits that can substantially improve patient care. Doctors would be able to review exams on their workstations using a user-friendly review environment. They would be able to annotate exams, run image processing programs on the images to vary image contrast to help them better analyze the images, view and compare multiple images at the same time, and view images concurrently with colleagues over great distances.

A typical Picture Archival and Communication System[1] (PACS) consists of a number of different types of computers interconnected by a communication network; at any point in time, there may be many different tasks competing for the system resources.[Pavicic (8), 370] The behavior of such a complex system is not intuitive. In the case of IMACS, a number of factors need to be taken into account in deciding how to proceed in connecting the HP SONOS ultrasound machines to create a reliable, available, high-speed network to support cardiology's imaging needs. Network options abound for choosing the physical medium, the topology, and the protocols to use. In addition, there are also many related technologies which will have an effect on an echo lab, such as: the storage facilities, and user/server workstations on the network.

1.1.1 Radiology PACS v. Cardiology IMACS

Much PACS research has been done, but mostly on systems for dealing with radiology data. Little research has been done on such systems for cardiology units in a hospital. The

1. An outdated definition for PACS is Picture Archiving and Control System.

needs of a radiology department and a cardiology department are significantly different because of the type of data used for the medical diagnoses. Radiologists use still shots—a snapshot of the organ being imaged. Radiologists do not need to know how the kidney looks from one second to the next; cardiologists, however, need to know how the heart is changing over time, and need to watch a sequence of image frames that show the heart in motion. Cardiology imaging involves storing potentially very long sequences of images because not only is it important for cardiologists to see the beating heart, but they must see it from a number of different views. Also, some cardiology imaging modalities such as ultrasound, can vary the processing and presentation of images to allow better visualization of wall motion, tissue texture, blood flow, or blood turbulence. Many of these modalities, such as Color Flow, Doppler, Two-D, and M-Mode are usually recorded during a typical exam. The typical radiology PACS system cannot provide this kind of real-time high data service.

Since radiology PACS systems only deal in single images, a busy network does not pose as much of a problem to a radiology machine as it does to an ultrasound machine. A computed radiography (CR) image (2048 x 2048 x 10 bits) is composed of 8 megabytes of digital data. If the network is busy in trying to save a radiological image, then the machine can easily save it in a local buffer. If a cardiology network is busy, then the sonographer will be unable to save ultrasound images while the network is busy. To allow the sonographer to record whenever he or she wants requires that the network be able to absorb the image data in real time, as it is streaming out of the ultrasound machine.

1.2 Project Definition

A software model was developed to allow the hypothetical examination of how various IMACS configuration options interact. Apart from performance prediction, a computer simulation of an echo lab network could also be used for hypothesis testing to answer "what-if" questions in the further design of the cardiology network. The model should be able to shed light on how the IMACS system would behave with different local buffering capacities, storage transfer speeds, network medium speeds, and sonographer usage patterns.

The project consists of two integrated parts: characterizing the traffic that can be expected to flow across the echo lab network, and simulating the effect of this traffic using a discrete even simulation of the whole IMACS network. A software model of a network consisting of a transfer medium, a variable number of SONOS ultrasound machines, a storage system, and a variable number of reviewstations, was developed using the communication simulation/analysis package OPNET[1]. OPNET is a network modeling package which allows the simulation and analysis of different network topologies, protocols, and algorithms. Models were developed and/or parameterized to represent: two high speed networks (Fast Ethernet and 100VG-AnyLan); two disk transfer rates (normal SCSI-2 and fast-wide SCSI-2); and two disk transfer sizes (4KB and 512KB).

In order to characterize the expected workload, research was conducted to understand the SONOS ultrasound machine usage patterns. Since this image traffic will dominate the network, it was important to characterize it and understand its effect on network performance. The storage traffic model was derived from clinical videotapes of the current SONOS usage patterns during typical cardiac examinations. The characterization of the network traffic only involves such a detailed analysis on the storage side. The traffic on the review side can be modeled as bulk transfers that, relative to the storage traffic, occur rarely. Under this assumption, the reviewstations can be equipped with enough buffer space to hold a few exams, assuring that exams can be loaded into the workstation memory during lulls in the storage network traffic.

1. OPNET is registered trademark of MIL 3, Inc.

1.3 Summary

This project simulates a small Image Management and Communication system for cardiac ultrasound, examining how different system configurations affect the ultrasound buffers, delay times, and the time it takes the reviewstations to receive exams.

The rest of the paper is divided up as follows:

- Chapter 2 discusses previous work involving simulations of image management networks.
- Chapter 3 covers the research into the current sonographer usage patterns and the network load characterization.
- Chapter 4 presents a detailed overview of the simulation model, including development and validation.
- Chapter 5 presents the simulation results showing that the network medium is not as important as the block size of the transfer to the disk.
- Chapter 6 concludes by discussing the results in general, and offering ideas for future work.

Chapter 2

Previous Research

2.1 Techniques of Performance Evaluation

IMACS are complex systems in which performance can be limited by any one of several subsystems, including the network, memory, or CPU. [Saulnier (11), 178] Seeing that the issue of performance is not at all intuitive, and that there are many opportunities in designing an IMACS network for something to go wrong, many image management network designers have turned to performance evaluation. Performance evaluation attempts to answer questions about characteristics such as throughput, latency, and system bottlenecks.

Three techniques for performance evaluation are analytical modeling, simulation, and measurement. [Jain (12), 30] Analytical modeling involves creating an on-paper model using a mathematical approach such as queueing theory. Simulation typically involves creating a discrete event model on a computer that attempts to mimic how the system to be modeled behaves. Measurement involves running benchmark tests on an already existing system to find the metrics such as system throughput.

Computer simulation is an effective way to analyze the performance of a complex system. Computer simulations can be designed to mimic the exact system in conception, at whatever level of detail is appropriate for the information desired. This chapter presents a little background on network simulation and analysis to give a framework for understanding the IMACS simulation models which follow in Chapter 4.

Simulation is often used as a means to understand or analyze some system, often when it is impossible to deal directly with the system because it may not yet exist, or the cost of the studying the system directly may be too high, or is impractical due to time restrictions.

[Garzia(8), 1] Since simulation involves abstracting away from the details of a system to capture only what is important, a simulation model is often much easier to understand and modify than the real system when trying to vary the system's behavior. Very complex systems may be adequately captured by simple models that describe some key behaviors of interest.

Another benefit of conducting simulations is that an understanding can be gained of how the system works, or how it should work. Even before a simulation model is completed, flaws in the real system design may be discovered during the model development period.

The flip side is that simulations are extremely difficult to verify. True verification rests in actually measuring the items of interests in the real system and comparing to what the simulation predicts. Of course, actual measurement has its problems as well, since it is extremely likely to alter the system being measured simply by measuring it.

2.2 Techniques of Simulation

Simulations can be classified into one of three types: continuous time, discrete time, and discrete event. [Garzia (8), 2] Continuous time simulations are based on ordinary differential equations, and discrete time simulations are based on difference equations. Discrete event simulations, however, view the system as a group of interacting objects, where any system changes are triggered by events that happen at certain times.

Discrete event systems cannot be modeled by standard difference equations or differential equations since the evolution of the system is not simply controlled by the passage of time, but by the interaction of the objects and the events they trigger. [Garzia (8), 2] For discrete event simulations, a computer program must be used to keep track of the objects in the simulation, and the interactions between them.

In the course of the simulation, an *event-list* is updated with new events when objects trigger interrupts. For example, in sending a packet from one node to another, the time for the packet's trip across the network is calculated to know when the packet will arrive at the destination. The event of the packet arriving at the destination at the specified time is enterend into the event list. There is no real flow of time since the simulation is not continuous; instead the simulation clock is merely advanced the amount of time betwen events on the list.

2.3 PACS Simulations

A number of PACS systems have been simulated using computer models. The goals of most of the studies were for performance prediction in order to aid in the design of the system, or to gauge the value of the system.

In the Netherlands, at BAZIS (Central Development and Support Group Hospital Information System), modelling and simulation were extensively used to investigate and predict the performance of future image information systems. BAZIS developed the modelling environment and simulation package MIRACLES (Medical Image Representation, Archiving and Communication Learned from Extensive Simulation) to simulate various PACS designs [Stut(16), 423]. Miracles is a simply a framework written in the simulation language Prosim, but is powerful enough to simulate various PACS designs.

BAZIS initially used the package to simulate hypothetical PACS systems. In one example, a three-level storage hierarchy system was proposed for a digital thorax section of the Leiden University Hospital radiology department. The analysts used the simulation to find the effect of the buffer configuration on the system's performance. BAZIS simulated 6 different buffer configurations: the number of (Winchester) disk units was either 2 or 3, and the data transfer rate could be 0.5, 1.0, or 1.5 MB/s. Simulation showed that during the description process, the time needed to display a set of 4 images was hardly affected by the buffer configuration chosen; it varied from 4 to 6 seconds. However, the time needed to display the patient's pictorial index (12 MB) during the selection process was heavily influenced by the buffer's transfer rate: 15 seconds at 1.5 MB/s, and 45 seconds at 0.5 MB/s. [Stut(16), 424]]

BAZIS used the modelling method to simulate PACS systems in Dutch hospitals, and later, the model was used to simulate the Geneva PACS to aid in software development of a hospital wide integrated Image Management System. [Stut (25), 37] The Department of Computer Science and Operations Research at North Dakota State University developed a software simulator to evaluate the performance of the Mayo/IBM PACS in a variety of different designs. The goal of the simulator was to explore performance related issues prior to the design installation, so that information about potential bottlenecks, resource utilization, and response times could be obtained while changes were still easy to make. [Pavicic (8), 370] A complex simulator was developed to accommodate changes to the network configuration and device characteristics using a block oriented approach.

The IBM Federal Systems Division also used computer methods to determine the configuration needs of a PACS system. Using a queueing model approach, IBM's Network Systems Laboratory looked into whether CSMA/CD or token ring would make a better network choice for a radiology PACS system. [Lawrence (6), 730] A 10 Mbps Ethernet was compared against a 4 Mbps token ring, in simulation, and response time results showed that the LAN media was a less severe constraint compared to the internal architecture and programs of the workstations. [Lawrence (6), 729] Basically, IBM says the speed of the transmission medium is not the problem in building a high speed PACS system. The Department of Radiological Sciences at UCLA found, in contrast, that their PACS throughput was limited by three major factors: (1) low-speed data interface used in the radiologic imaging devices; (2) competition for systems processing time among the PACS processes; and (3) network degradation caused by the heavy network traffic. [Wong (21), 252]

A Canadian simulation study conducted at the University of Alberta looked into the viability of Ethernet for a radiology PACS system. [Davis (2), 740] Performance measures such as response times, channel utilization and number of collisions were obtained from the simulation of a distributed imaging system for storing and retrieving digitally formatted medical images. The simulation study found that using an Ethernet as the transmission medium did not give reasonable response times of less than 5 seconds if the number of users was high. Only three users with 4096x4096 pixel images could be supported on a 10 Mbps Ethernet, and 18 users if the image size was 2048x2048. [Davis (2), 740]

Many of the above PACS simulators and models were developed from scratch using special languages that support queueing analysis or discrete event simulation. Another approach to building a simulation model, and the same approach undertaken in this simulation of an ultrasound IMACS, is to use a pre-existing communications analysis package. The Medical Imaging Division of the Department of Radiological Sciences at the UCLA School of Medicine used the Block Oriented Network Simulator (BONeS)[1] to perform discrete, event-driven Monte Carlo simulations in capacity planning, trade-off analysis and evaluation of alternate architectures for a high-speed, high-resolution teleradiology project. [Stewart (15), 2]

The software model developed by UCLA with BONeS successfully answered certain design questions concerning the system. The simulation found the system's reasonable delay service limit. UCLA also learned that compressed video teleconferencing could be run simultaneously with image data transfer without impacting the network performance,

1. BONeS is a registered trademark of Comdisco, Foster City, CA.

and that upgrading to a higher bandwidth WAN (Wide Area Network) link would not improve the system performance. [Stewart (15), 10]

2.4 Traffic Workload Analysis

The type of network traffic workload used in the network simulation model is very important to the reliability of the study. Many PACS studies simply assume a Poisson distribution for the storage and view requests, and since most communication analysis packages provide for these distributions, assuming a standard distribution is a fast way to obtain some performance information about the network without looking into the complexities that make up the actual network traffic patterns. The selection of a workload model is a crucial part of any IMACS simulation and should be representative of the real application. [Saulnier (11), 183] GE Corporate Research and Development used the BONeS package to see how the performance of a radiology PACS system (over token ring and Ethernet) varies with workload. GE found that using the standard Poisson distribution to evaluate PACS performance led to overly optimistic results. [Saulnier (11), 178]

Siemens and Loral Western Development Labs also used a communication analysis package to model and simulate a large PACS system. [Anderson (1), 577] The modeling tools LANNET II.5 and NETWORK II.5[1] were used to hierarchically model the Medical Diagnostic Imaging Support System (MDIS), a complete digital computer based system for storage, retrieval, and viewing of radiological images. The Siemens study made some interesting assumptions about the network traffic:

> The interarrival time distributions are derived for the model from the overall image traffic rates in terms of new images per year. These rates are translated into daily rates in terms of CR equivalent images, and then using the 5 Busy Hour Day concept, into rates in CR equivalent images per minute. The 5 Busy Hour Day concept is an approach that considers the worst case scenario for system loading, by using the assumption that the peak hour's loading represents 20% of the entire day's loading. [Anderson (1), 582]

1. LANNET II.5 and NETWORK II.5 are registered trademarks of CACI Products Company.

Siemens basically used an average worst case interarrival rate to test the performance of their PACS system. This approach is good for knowing how the system behaves during peak traffic times, but fails to give an accurate description of how the system will behave in the normal case.

Although workload characterization is important in developing an Image Management and Communication System for cardiac ultrasound, to date, not much work has been carried out on analyzing the ultrasound usage and potential digital workload. From a survey of cardiologists [Kung (13), 82] it was found that physicians say they store an average of 10.1 minutes of video on a VHS tape. However, this is a perceived, instead of measured amount of average storage; so it is necessary to actually quantify how the ultrasound machines are used. The survey also shows that cardiologists claim to store an average of 6.2 views per exam. [Kung (13), 82] However, the survey does not give much information on the breakup of a typical exam.

2.5 Summary

Computer simulation may be an effective technique for performance evaluation of a network especially when the network is hypothetical and cannot be measured. Simulation, and in particular discrete event simulation, has proven useful in other studies of hospital imaging systems.

Imaging systems for cardiac imaging is a relatively new area, and not much work has been done before on networks to support the high data needs of cardiology labs. Radiology, in contrast, has seen substantial work in Picture Archiving and Communication Systems to support its needs. Moreover, no work has been done in characterizing the expected workload in an echolab from empirical data.

Chapter 3

The Workload Model

3.1 Workload Characterization in General

The workload is the most crucial part of any performance evaluation project since it is possible to reach misleading conclusions if the workload is not properly selected. [Jain (12), 60] In terms of a network with clients and servers, the workload may be described in a number of different ways, such as: the most frequent request, the worst case request, the average request, a mathematical statement describing the mean arrival rate of requests, and an actual time-stamped trace of the requests.

Since discrete event simulations of the network are being conducted, it is possible to simulate the actual usage patterns of the ultrasound machines without having to abstract this storage workload to a function of the mean arrival rate. Basing the simulation workload on the expected usage of the ultrasound machines gives a more representative input for the simulations, showing the kind of loading the network can expect in the average case.

The important workload to determine is the average case workload. The expected traffic on the echolab's image management network is a mix of image information streaming out of the ultrasound machines during exams, and the requests generated by the review workstations when the cardiologists want to view and manipulate the images. The flow of information on the network is limited to the transfer of image data from the ultrasound machines to the file server (the storage traffic), the transfer of requests from the reviewstation to the fileserver, and the image transfer from the fileserver to the reviewstation.

3.2 Characterization of the Storage Traffic

The storage traffic model was derived from clinical videotapes of current HP SONOS ultrasound machine usage during typical cardiac examinations. Currently, ultrasound exams average about ten minutes of real-time video (total exam time less the amount of gap and freeze time) recorded intermittently as snippets of varying length over the 45 minute to 1 hour exam. For a nominal 80 beats per minute heartrate, this is about 800 beats of information. In actual use, however, much fewer than 800 beats are stored because the video tape is often used to record a prolonged view of a frozen, or still image. If the system were fully digital, it is assumed that this number would be reduced dramatically, mainly because loops of a restricted number of images per heartbeat could be used instead of long tape sequences. Moreover, still images would not be stored as a sequence of frozen images as they are on video tape, but simply as one image frame.

3.3 Data Collection

A direct way to characterize the pattern of writes to the digital storage, is to observe the actual usage of the ultrasound machines when the sonographers are recording exams. The image data the sonographers currently save on the VHS videotapes closely correlates to what a digital echolab can expect to save to disk.

3.3.1 Assumptions

As previously discussed, the amount of digital image data that needs to be reviewed will be much less than what is currently being saved. Nonetheless, people are slow to change. In the near term sonographers will continue to save as much as they were used to saving before. A digital echolab must be prepared to handle the amount of data that is currently being saved in the analog echolab.

3.3.2 Method

A number of clinical videotapes of pediatric and adult cardiac ultrasound exams from the DHMC (Dartmouth Hospital Medical Center) and the Elliot Hospital were used in gathering the data. Because of the different nature of adult and pediatric care, it is important to see what differences exist in the characteristics of the ultrasound exams for each group. Data was taken from seven adult, and five pediatric ultrasound exams. The average adult exam studied was 16 minutes 4 seconds long, whereas the average pediatric exam turned out to be 32 minutes 11 seconds long.

The collection method involved viewing the videotapes with a VCR and recording the span lengths of the different modes using the time kept by the ultrasound machine system clock (as displayed in the images during the exam).

Spreadsheets in Lotus 123 and Xess were used to store the raw time span data from the exams from the Elliot and Dartmouth hospitals. Later a C program, **pmf,** was written to give a frequency distribution of the data for each mode (2D, Color Flow, Freeze, M-mode, 2D Doppler, CF Doppler, and Gap). Each frequency distribution generated by pmf is used as a probability mass function to generate the appropriate span lengths for each mode.

pmf was used to find the frequencies of times spans in "buckets", ranges in seconds such as (1-4, 5-8, 9-12, 13-20, 21-32, 33-44, 45-60, 61-76, etc.) in order to know what percentage of the time a certain length of span could be expected. The buckets were chosen mainly to bunch together longer segments, because they occurred much less frequently than did the shorter segments, and to simplify coding of the SONOS model.

3.4 Exam Analysis

Two main characteristics of the workload were studied: how often the different modes (2D, CF, Freeze, M-mode, 2D Doppler, CF Doppler) occurred, and for each mode, what span lengths could be expected.

3.4.1 Data Rates

Although each mode is medically important and distinct from the other modes, they can be abstracted based on the amount of data flowing out of the ultrasound machine during each mode. The data rates are based on: an image size of 512 pixels on a side where two dimensional images only use one byte per pixel, but color flow images use two bytes per pixel. The ultrasound machines can output 30 fps with 3x lossless compression when taking two dimensional data, 15 fps with 5x lossless compression when taking color flow data, and 0.467 fps when taking doppler or m-mode data.[1] The figure below, Figure 3.1, gives a comparison of the different data rates.

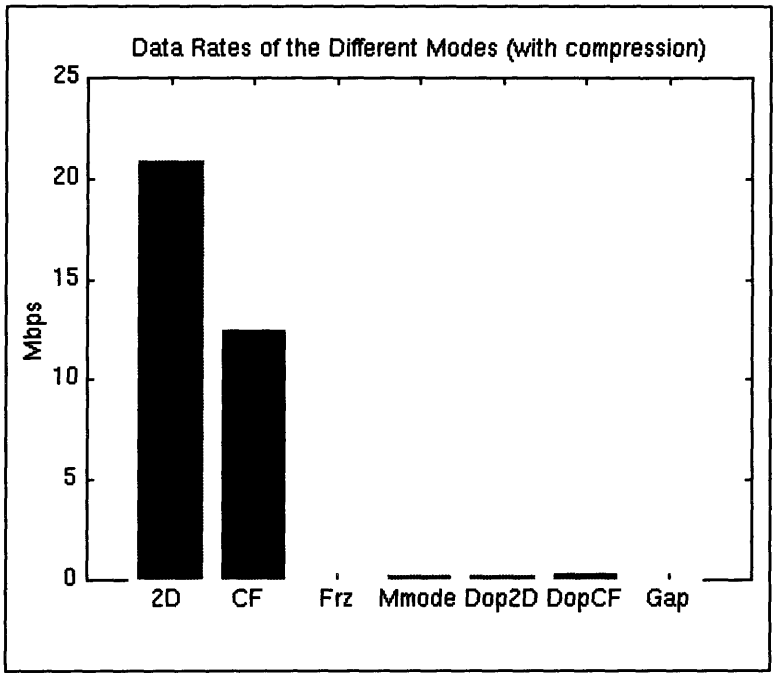

Figure 3.1: A comparison of the data rates of the modes. 2D generates 20.97 Mbps. CF generates 12.58 Mbps. Freeze generates only one frame, and then nothing for the duration of the still shot. Mmode and Doppler both generate 291.29 Kbps. CF Doppler generates 349.55 Kbps. No data is generated during a gap.

The next graph, Figure 3.2, shows the breakdown of the adult and pediatric exams by percent of data. The bulk of the exams is made up by two dimensional data, and the rest is made up by color flow data. Although medically important, the m-mode and the doppler

1. From multiple conversations with Barry Hunt and Grace Judy, both engineers at HP.

modes fail to contribute significantly to the overall amount of data generated during an exam.

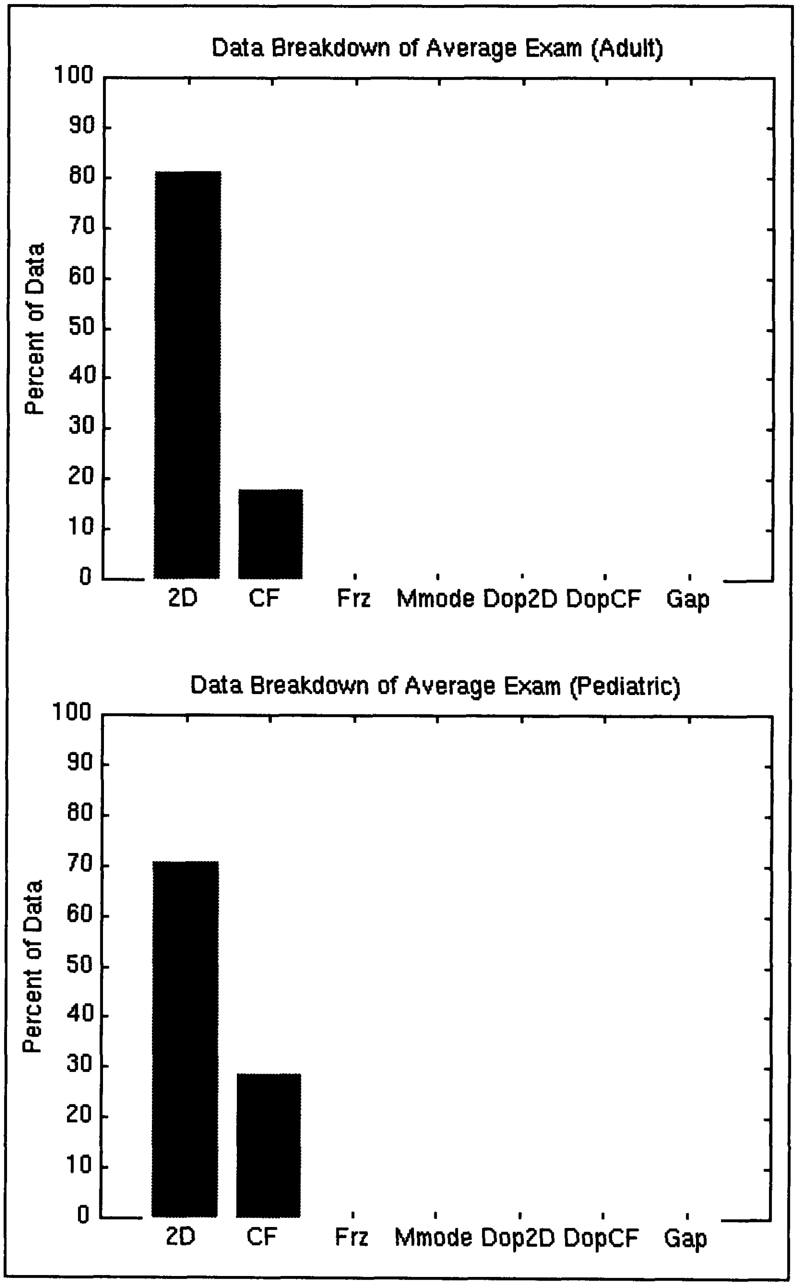

Figure 3.2: The breakdowns by percent of data of the average adult and pediatric ultrasound exams.

3.4.2 Frequency of Modes

The next graph, Figure 3.3, shows the percentage breakdown by time of the different modes in an average adult exam. The most striking thing about the graph is that it shows

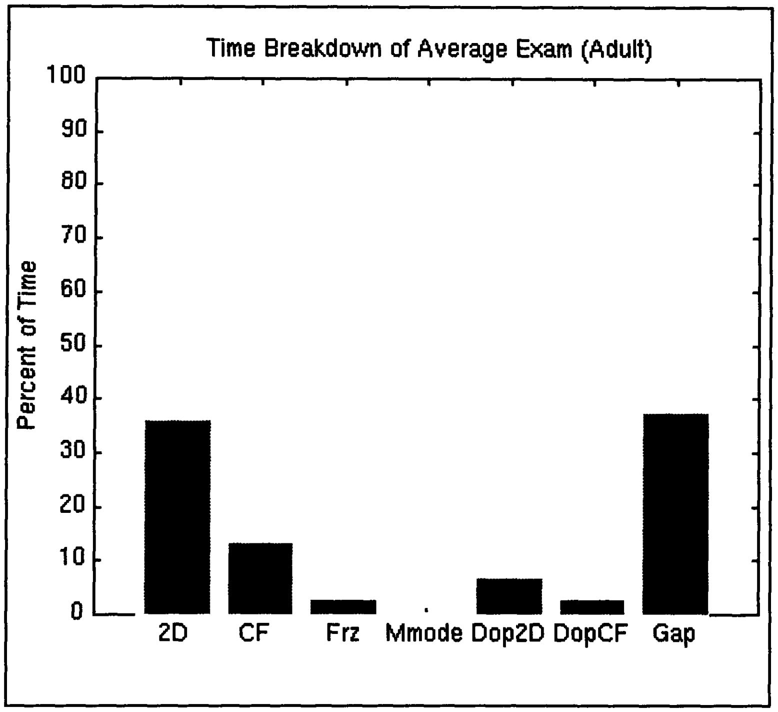

Figure 3.3: The percentage breakdown of the different modalities in the average adult ultrasound exam.

the importance of gap time. In the average adult exam, 37.4% of the time, nothing was being saved. Most of the gap time can be accounted for by the time it takes the sonographer to move the patient around, reapply the ultrasound gel to the transducer, and find the desired view of the heart (parasternal long axis, parasternal short axis, apical 4-chamber, apical 5-chamber, and subcostal).

The graph also shows that 2D (36.1%) and ColorFlow (13.5%) are the most used modes, with 2D Doppler (6.7%) a far third. This is important because it means that although the SONOS has modes such as M-mode (0.5%), 2D Doppler, CF Doppler (2.8%), where it is generating much less data than when in 2D or ColorFlow, those modes are hardly ever used. The impact is that although the SONOS is a device generating different amounts of data at different times, most of its time is devoted to either generating nothing or generating 2D data.

Another interesting point to note is the relative lack of stills or freezes. The storage savings that still shots could give a digital implementation is insignificant.

In comparison to the average adult exam, Figure 3.4, shows the average pediatric exam. The breakdown is similar to that of the adult exam; the only exception is that there

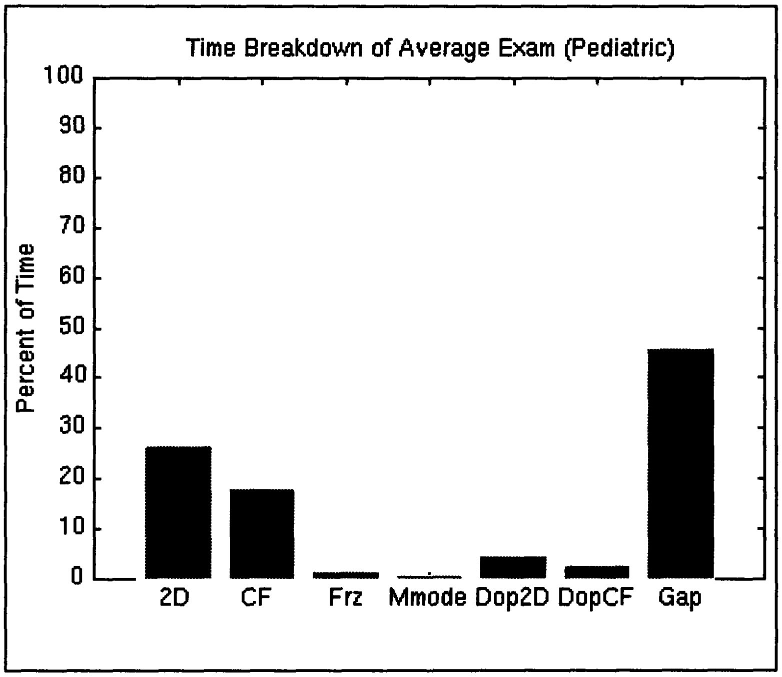

Figure 3.4: Percentage breakdown of the average pediatric exam.

is considerably more gap time in the pediatric exam. During the average pediatric exam, 46% of the time, no exam data is being saved. The larger amount of gap time is most likely due to the fact that adults are much easier to control as patients than children or infants. Also because of the smaller size of the organs in a pediatric exam, it may be much more difficult for the sonographer to locate the desired views during an exam.

3.4.3 Span Length Frequency

The other important element in characterizing the SONOS traffic generation is to find how long the bursts of data are between the gaps when the machines are not generating any data. This involves finding out how long the machine is generating when in a data generation mode such as 2D or CF, and how long the machine is sitting idle when in a gap between bursts.

The next graph, Figure 3.5, shows the percentage breakdown of the span lengths for 2D mode in the average adult ultrasound exam. The span lengths are shown in a bar graph

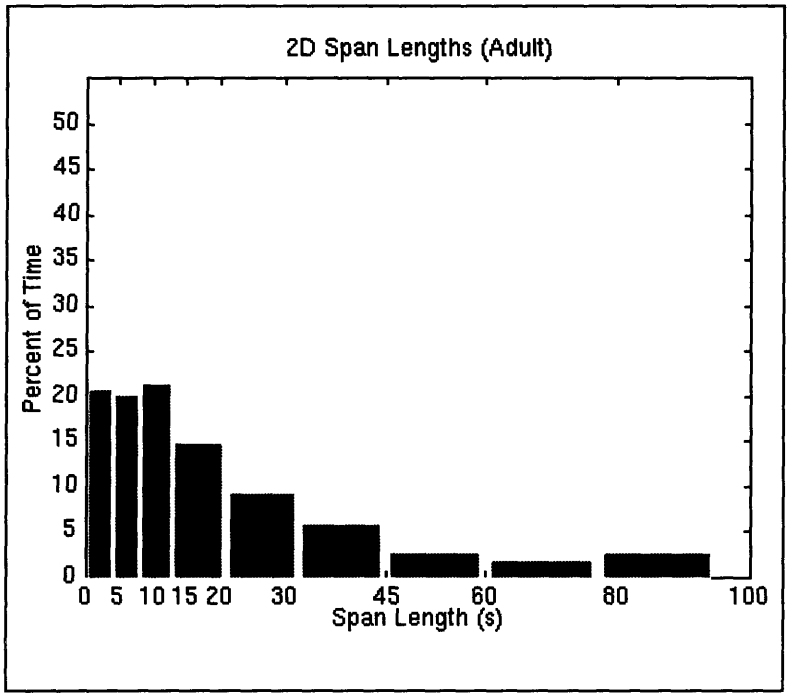

Figure 3.5: The span length distribution for 2D. (Adult)

where each bar is related to certain groupings of the span lengths. The bars in the graph are centered at: 2, 6, 10, 16, 26, 38, 52, 68, and 86. The histogram was created in this manner, instead of dividing the whole range equally, to show better separation with the shorter span lengths that occur very often, and not to obscure the fact that large span lengths do occur.

Similarly, Figure 3.6, shows the percentage breakdown of the span lengths when the ultrasound machine is generating color flow. In contrast to 2D imaging, color flow does not have as many short bursts less than 5 seconds as 2D does. The spans are generally in the 5-20 second range whereas the spans for 2D are generally in the 0-15 range. The chance of seeing an extremely long burst is also less likely when taking color flow data than 2D data.

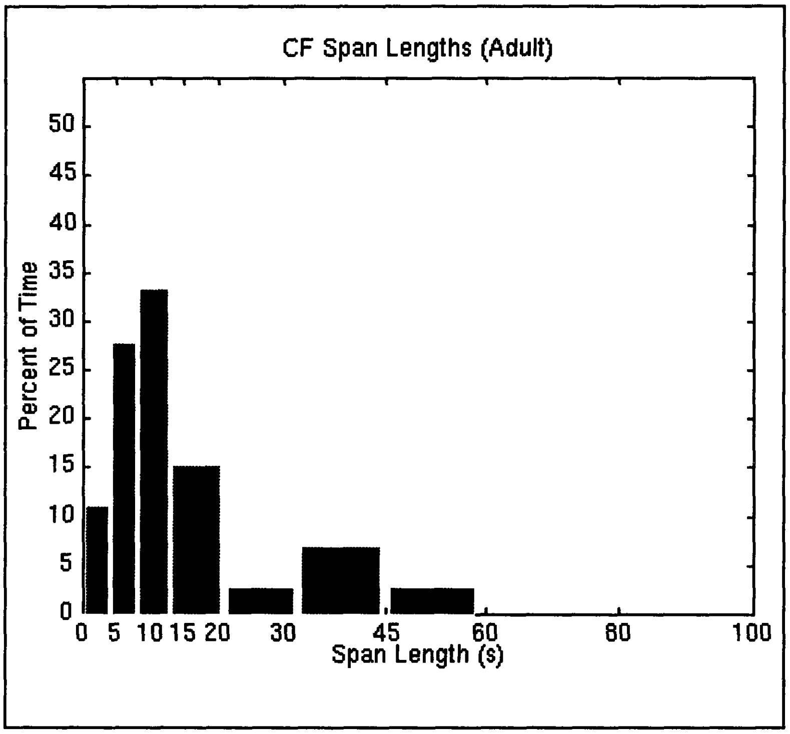

Figure 3.6: The average span lengths for color flow. (Adult)

Due to the infrequency of the doppler modes, the data for the 2D and color flow doppler spans is very noisy. It is interesting however, to look at the spans encountered in the studied exams just to see if anything can be gleaned from the graphs about how the doppler modes compare to the non-doppler modes.

Figure 3.7 shows the adult exam span lengths for 2D Doppler and Figure 3.8 (next page) shows the span lengths for CF Doppler. In contrast with the non-doppler modes,

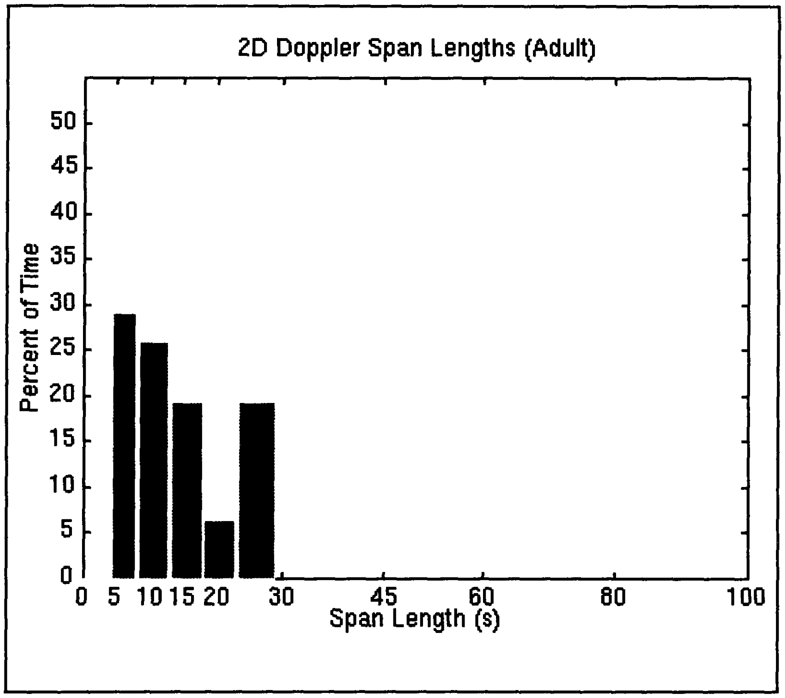

Figure 3.7: The span lengths for 2D Doppler. (Adult)

there are no very short span lengths. In fact, there are no span lengths under 5 seconds. The 2D Doppler spans lengths are typically between 5-17 seconds, and although some large span lengths do occur, they are not nearly as long as some of the 2D spans. The longest 2D Doppler span encountered in the studies was 38 seconds, while for 2D it was 114 seconds. Similarly, for color flow doppler, illustrated in Figure 3.8 below, the longest span encountered in the study was 31 seconds.

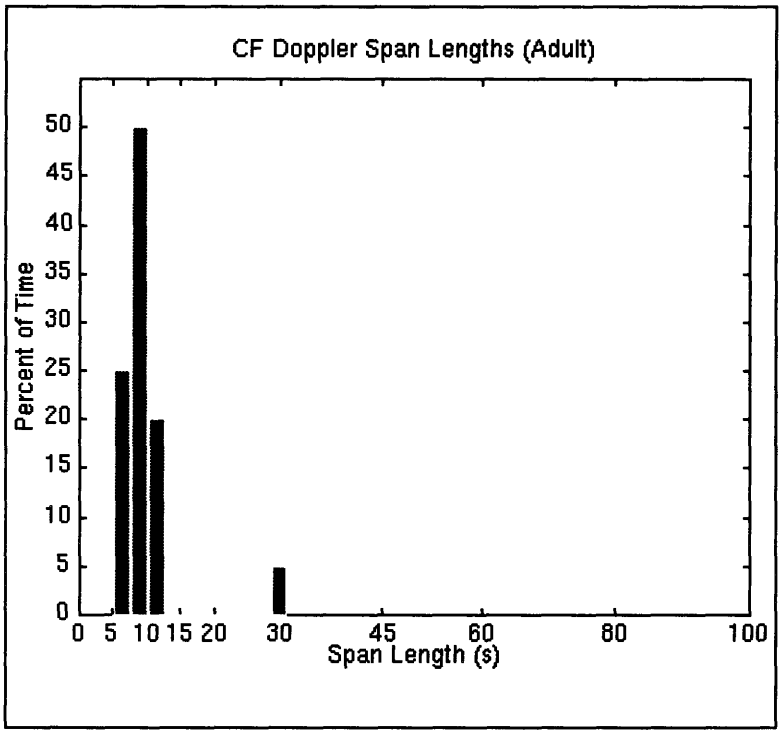

Figure 3.8: The span lengths for CF Doppler (Adult)

After having looked at the frequency and duration of the different data generation modes of the ultrasound machines, it is important to look at the time that the machine does not generate any data as well. The lack of data generation can therefore also be viewed as

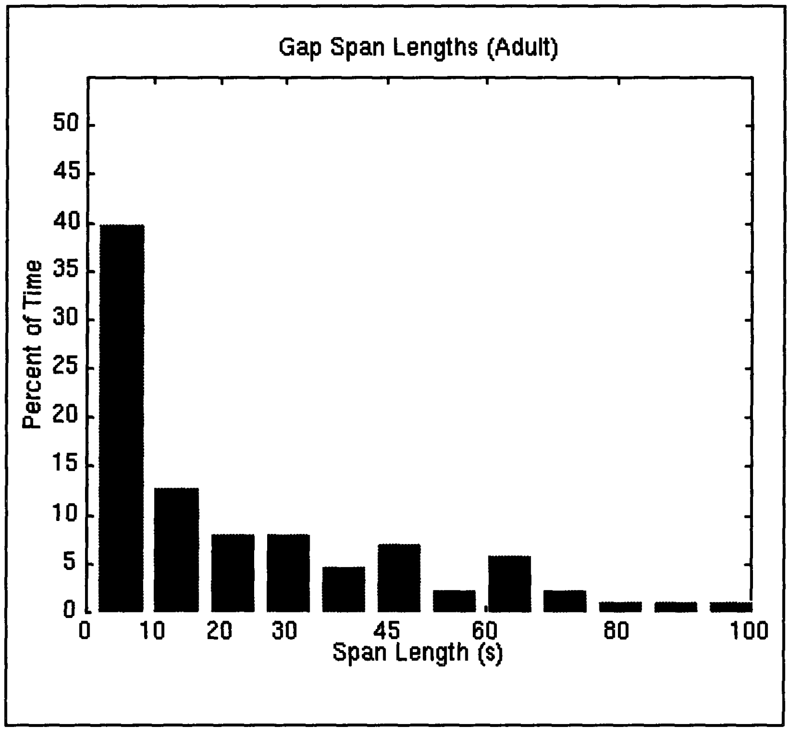

Figure 3.9: The lengths of the gaps in the average adult exam.

a mode that the machine could be in. The span lengths of the gaps encountered during the average adult exam are shown in Figure 3.9. The interesting thing to note from this graph is that almost all of the gaps are under a minute, although there are some very long gaps (the longest measured was almost 3 minutes.)

3.4.4 Pediatric Exams

Similar data was recorded for the pediatric ultrasound exams, and the usage patterns were found to be extremely similar to the patterns for the adult ultrasound exams. The graphs are presented in the appendix. One difference, already noted however, was that there is significantly more gap time in the pediatric exams. The span lengths of the gap time for the pediatric exams, shown in Figure 3.10, are slightly different from those encountered in the adult exams. The average pediatric exam has a higher percentage of

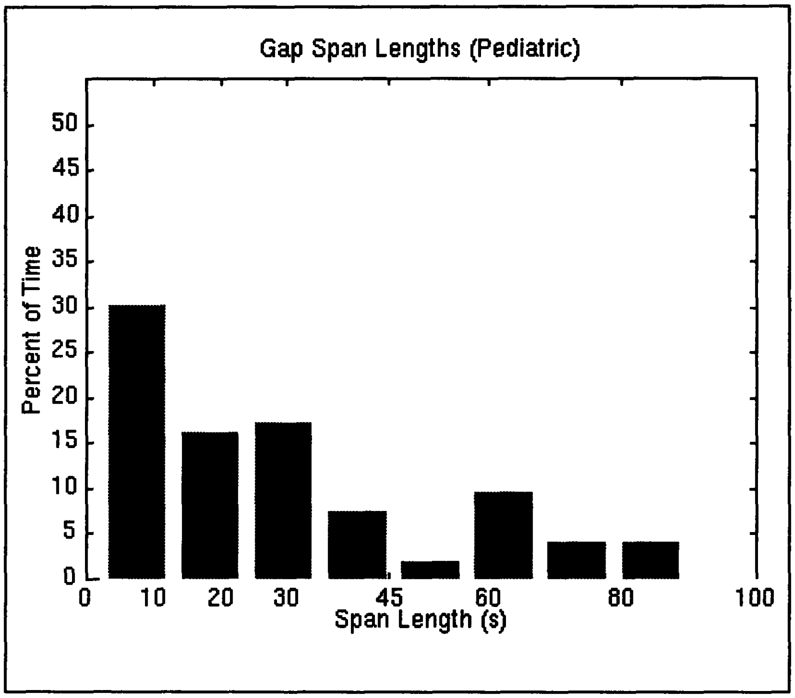

Figure 3.10: The spans lengths of the gaps in the average pediatric exam.

time spent in gaps between 20 and 40 seconds than does the average adult exam, although very long gaps are rare, like they are in the adult exam.

Another interesting difference between the adult and pediatric exams is the span lengths for the color flow segments taken. The color flow span lengths for the average

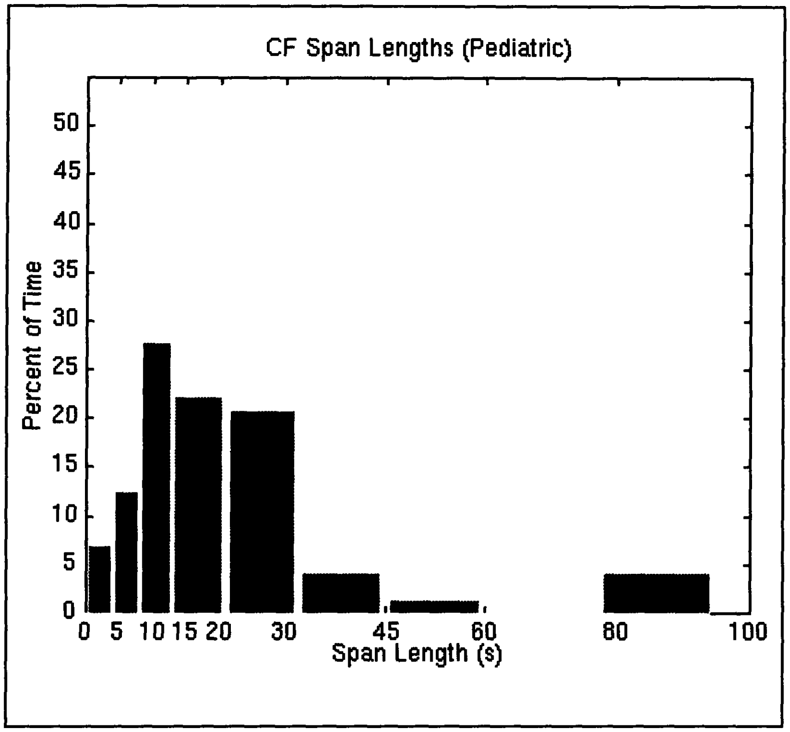

Figure 3.11: The color flow span lengths for the average pediatric exam.

pediatric exam are shown in Figure 3.11. The key difference between the graph of the adult span lengths for color flow (Figure 3.6) is that almost 22% of the span lengths for pediatric color flow are between 20 and 30 seconds, whereas that region of span is almost insignificant for the average adult exam. This means that the color flow segments of a pediatric exam are longer than the color flow segments of an adult exam.

3.5 Summary

Although there are some slight differences between the adult and pediatric exams, only the adult exam workload will be considered in the simulations. Generally, during any given minute, more data will be streaming out of an ultrasound machine during an adult exam than a pediatric exam. However, the total size of a pediatric exam is usually larger than an adult exam merely because the pediatric exams take about twice as long as an adult exam. Simulating adult exams, therefore, would present a more pessimistic workload to the network.

The clinical ultrasound exams of adult and pediatric patients studied have shown that for a significant portion of the exam time, the ultrasound machine is idle. For the average adult exam studied, idle gap time made up 37.4% of the exam. For the average pediatric exam, 46.0% of the time was spent idling. The rest of the time is basically split between two dimensional imaging and color flow imaging. The doppler modalities (color flow and two dimensional) and the M-mode modality are hardly used at all. The significance of the prevalence of gap time in the exams is that the ultrasound machines are not continuously streaming data onto the network, but instead have a bursty pattern.

Chapter 4

The Simulation Model

4.1 Modelling Framework

The simulation model of the Image Management and Communication system was developed using the block oriented communications package OPNET, illustrated in Figure 4.1. Any networked system to be simulated, ranging from a small LAN for an echocardiography lab to a Wide Area Network connecting university networks, can be viewed as a hierarchy of several models at different layers: network, node, and process. The highest level is the *network* model. This level deals with the network medium, and the devices attached to it. From the viewpoint of this layer, all the nodes on a network are black boxes. Examples of such network nodes are: workstations, fileservers, ultrasound machines, RAIDs, and bridges or routers to other networks. OPNET allows the placement and interconnection of network nodes on a grid. OPNET Version 2.5 allows point-to-point, and bus communication between nodes on the network.

Each node on the network in turn has its own *node* model to describe how it works as a black box sitting on the network. The node model describes the flow of information within the node, and where any processing may take place. Queues, processors, transmitters, and receivers are connected together with either simplex packet stream wires or simplex statistic wires. Packet streams are used for carrying the data packets from one node to another, whereas statistic wires are used by nodes to pass information about the state of the node itself independent of the actual data which the node is handing.

Whatever real work needs to be done on the data is done by the processor or processors in the node model. Each processor is described by a *process* model consisting of a state transition diagram and the C code to be executed upon entering or exiting each state.

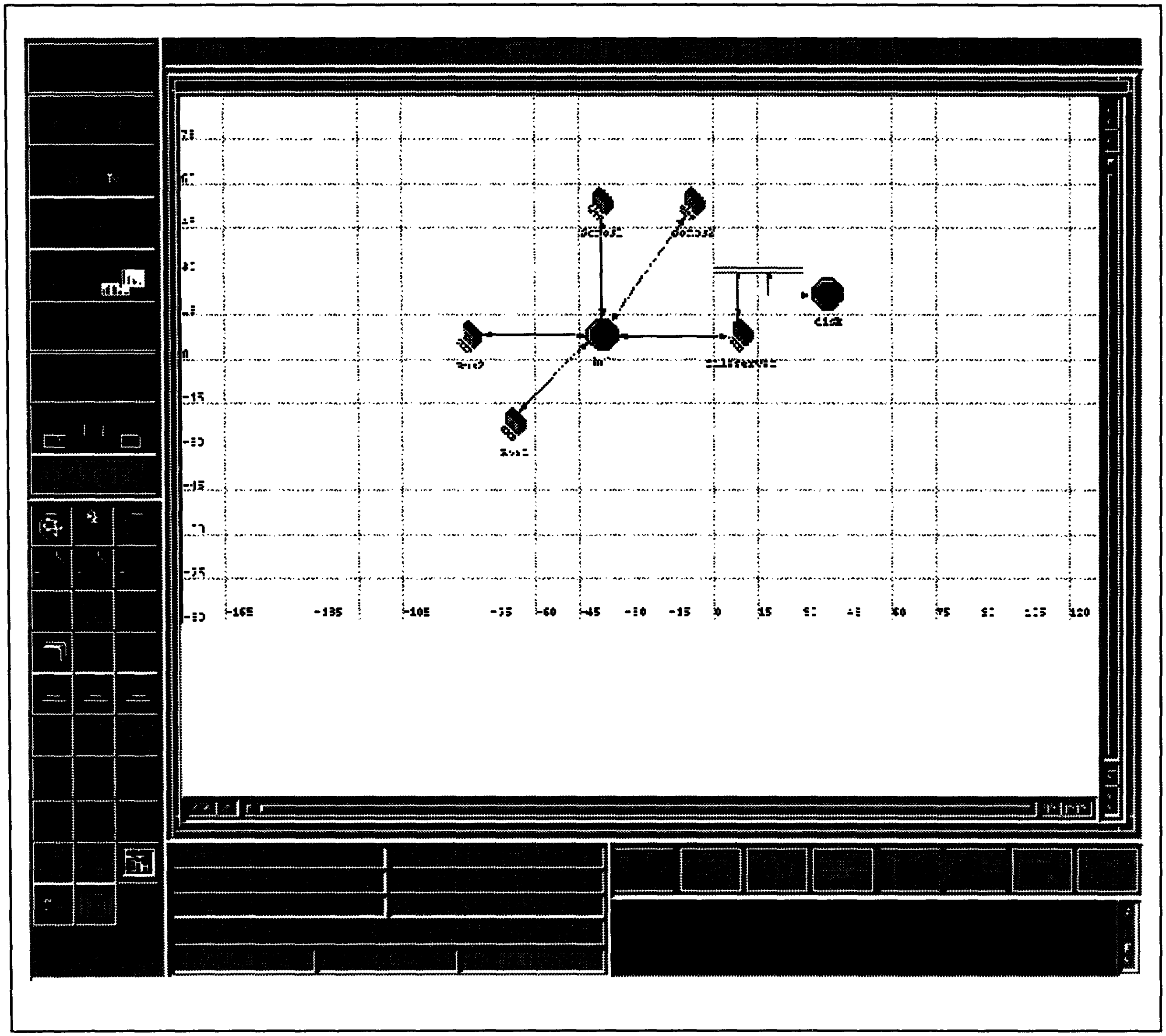

Figure 4.1: The OPtimized NETwork communications analysis tool.

An example of a process model state transition diagram and the code corresponding to one of its states is shown in Figure 4.2. The process model is event driven; as packets arrive on a packet-stream connected to a processor, they signal interrupts which start the execution of the process model. Transitions between the states in the state transition diagram are determined by the evaluation of conditions associated with the transition between any two states.

Because of the way OPNET works, a process model cannot simply execute its code continuously without giving up control to the simulation kernel. The OPNET simulation kernel allows each process model to run until it gives up control by transitioning into an unforced state. This allows the kernel to perform other functions necessary for the control of the simulation such as advancing the simulation clock, and passing control to other processes waiting to run. [MIL3 (3), Bpt-6]

OPNET allows for the run time analysis of any network model through the use of a separate *probe* model. Network probes can be attached to any of the nodes on the network to determine certain characteristics of individual elements in the node. Example performance measures are, for a queue, average queue size, and for a transmitter, bit throughput. In addition, OPNET allows the logging of certain user defined data; so code can be written to calculate the average end-to-end delay of packets in a system.

Using this layered approach to system modelling, models were developed for the SONOS ultrasound machine, fileserver, harddisk, and review work stations attached to the network. Because 100VG-AnyLan and Fast Ethernet both use the standard ethernet format, separate process models for all the nodes in each network model were not needed to simulate the creation of packets destined for these different types of networks. Separate node models, however, were required since the node model is where each device's method of network medium access is implemented. Different medium access methods are required

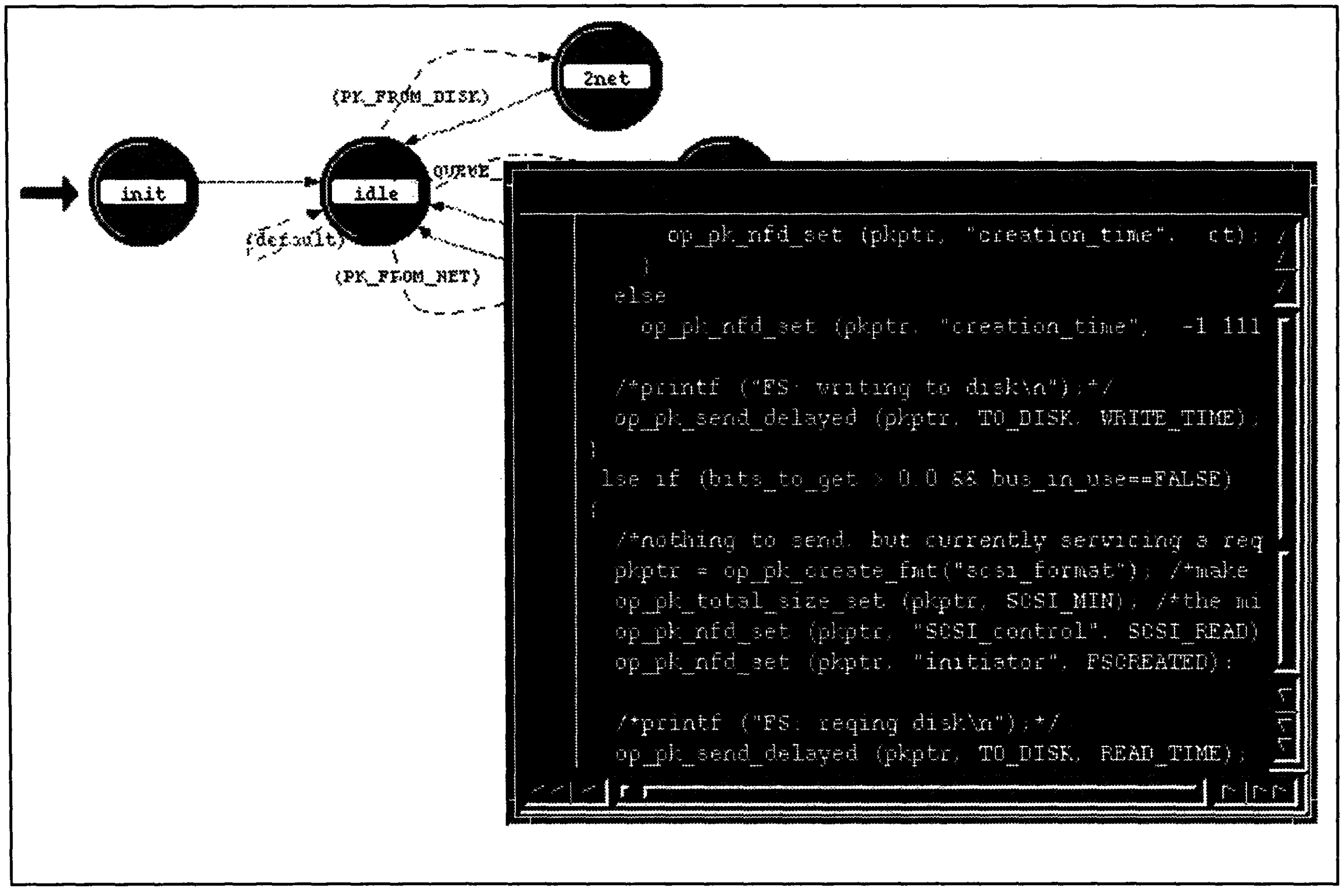

Figure 4.2: C code enter executives for the ops state in the fileserver process model.

depending on whether the network medium is the shared Fast Ethernet or the switched 100VG-AnyLan, since Fast Ethernet requires a tap to the shared bus, and 100VG-AnyLan requires a point-to-point connection.

4.2 The Echocardiography IMACS

The network level model of the Image Management and Communication System is shown below in Figure 4.3 and Figure 4.4. The network medium depicted in Figure 4.3 is the shared medium 100 Mbps Fast Ethernet which uses the CSMA/CD (Carrier Sense Multiple Access with Collision Detection) algorithm to control network access. 100VG-AnyLan, the switched network medium which uses the Demand Priority algorithm to control network access, is shown in Figure 4.4.

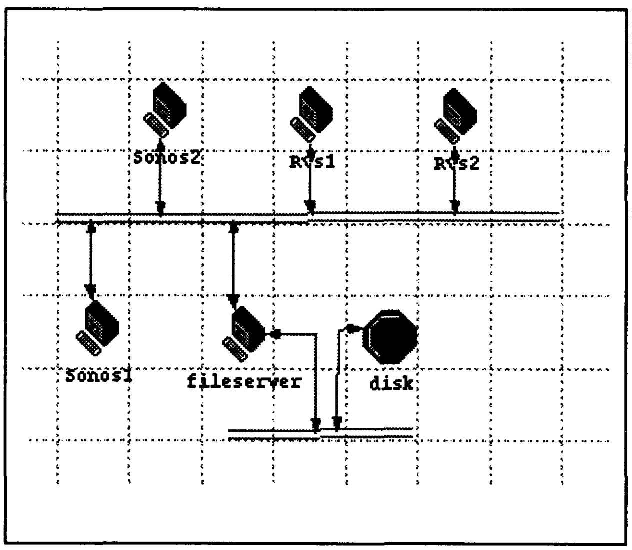

Figure 4.3: Echolab IMACS on Fast Ethernet (shared ethernet)

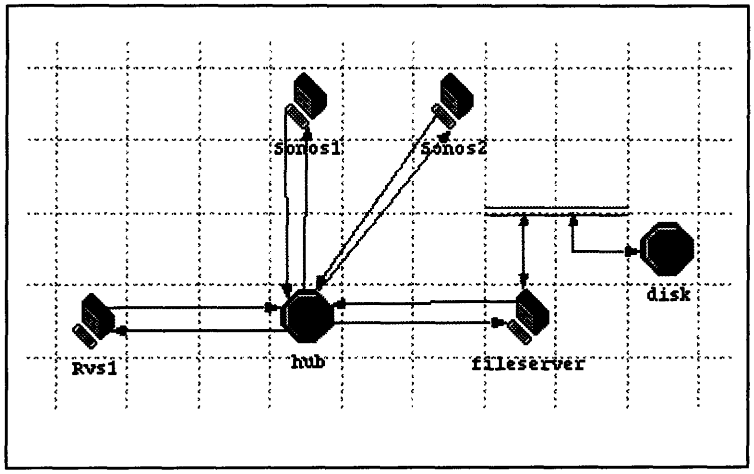

Figure 4.4: Echolab IMACS on 100VG-AnyLan (switched ethernet)

Both networks show an example echolab setup: two ultrasound machines, two reviewstations, and one fileserver, connected to a disk via a separate SCSI (Small Computer System Interface) bus connection, make up a small LAN (Local Area Network). Typically, the number of ultrasounds machines and reviewstations may vary slightly, but since the IMACS is intended to serve in a small echocardiography lab, the number of devices on the LAN will be limited.

4.3 The Fast Ethernet Model

100Base-T, also known as Fast Ethernet, 100Base-TX, and 100Base-T4, is one of two emerging standards for 100Mbps Ethernet. [Thomas-Conrad Corp. (26), 3] Fast Ethernet consists of a shared bus accessed by stations using the CSMA/CD (Carrier Sense Multiple Access/ Collision Detect) access method. The CSMA/CD algorithm allows multiple stations to share the same bus in a manner similar to what happens in cocktail parties where each guest waits for a lull in the conversation before blurting out what they have to say. If two guests start to talk at the same time, then they both apologize and back down to wait for another lull. Ethernet workstations, however, in the case of a collision back down for a random amount of time to reduce the chance of having the same two workstations retry after the same amount of time, and collide again.

The model of Fast Ethernet used here is OPNET's Ethernet model (as discussed in *The Opnet Example Models*) with a change of the bus data rate from 10 Mbps to 100 Mbps. The simple change provides for a decent model of Fast Ethernet, since Fast Ethernet is essentially the same as Ethernet, only ten times faster. [Farallon (6), 2] In fact, 100BaseT, or Fast Ethernet, retains the current architecture and CSMA/CD protocol of standard Ethernet, but speeds it up to 100Mbps. [Delaney (5), 6]

Figure 4.5 depicts the OPNET model of the ethernet medium access method at the node level. The **mac** processor-queue accepts packets from a higher level source (not shown in

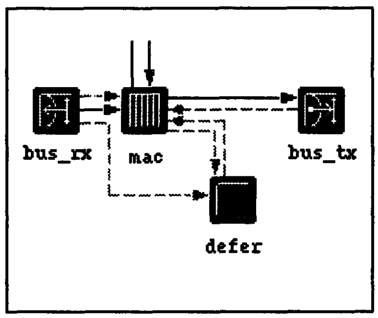

Figure 4.5: The Fast Ethernet MAC

the figure) and encapsulates them in ethernet frames. **mac** also accepts ethernet frames from the bus receiver **bus_rx** and extracts the data portion to forward on to a higher level sink process (also not shown.) The **defer** processor monitors the state of the bus and communicates to the **mac** processor-queue over a statistic wire whether the bus is free for the node to attempt to transmit. The **mac** processor also handles the random backoff if a collision is detected.

4.4 The 100VG-AnyLan Model

100VG-AnyLan is a switched ethernet-based network technology developed by Compaq, Compex, Hewlett-Packard, KTI Networks, Racore Computer Products, Texas Instruments and Thomas-Conrad. 100VG-AnyLan allows for a 100 Mbps data transfer rate using the same frame definitions that Ethernet and Fast Ethernet use. [Hewlett-Packard (9), 1] In contrast to the shared 10 Mbps Ethernet and 100 Mbps Fast Ethernet, there are no collisions because 100VG-AnyLan dispenses with CSMA/CD in favor of a *Demand Priority*, a new protocol at the Media Access Control, or MAC, layer of the network developed by Hewlett-Packard to support video and voice traffic. [Delaney (5), 6]

4.4.1 The 100VG-AnyLan Hub Node Model

The node model of the switched ethernet 100VG-AnyLan Hub is shown in Figure 4.6. The hub consists of a number of point-to-point transceiver pairs, and the processor **hub**. The transceiver pairs act as the hub's network ports which are connected directly to the stations that sit on the 100VG-AnyLan network. Since the Image Management and Communication Systems to be modeled are typically small, the hub model shown below only has six ports, numbered 0 through 5 in the figure. The corresponding receivers and transmitters are set to receive and send data at 100 Mbps.

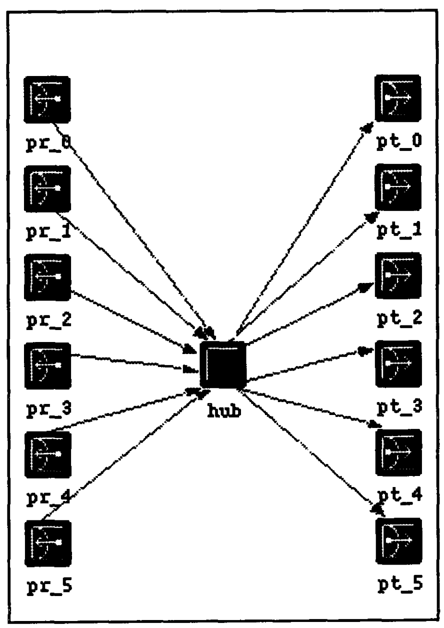

Figure 4.6: The 100VG-AnyLan Hub

100VG-AnyLan uses a centrally controlled access method called *Demand Priority.* Demand Priority is a simple, deterministic network access method in which nodes issue a request to the hub to send one packet on the network. [Hewlett-Packard (9), 2] Requests can be marked as either normal priority or high priority. High priority requests are sent before normal priority requests, so critical applications that need the bandwidth can get it. However, to prevent starvation, each normal priority request is timestamped and if not serviced after an established maximum amount of time, is promoted to high priority.

The hub continually scans its ports in a round-robin fashion, so that each port gets serviced in turn if it has any packets to send. In contrast to the medium access method CSMA/CD used by Ethernet and Fast Ethernet, every station on a 100VG-AnyLan hub is guaranteed to get its turn to send a packet. Since the hub cycles through the ports and handles the packet transfers if required, there are no collisions. In a 100VG-AnyLan switched network, if nothing is being transmitted on the network, it is because none of the attached stations have anything to send. In contrast, on a shared Ethernet, there may be time when nothing is being transmitted on the network because stations have collided, and are backed down waiting to retransmit.

4.4.2 The 100VG-AnyLan Hub Process Model

The simulation model does not continually scan its ports. To model the real hub with a 30MHz clock checking its ports, the hub model hub would need to schedule a self-interrupt every 33.3 nanoseconds of simulation time. At every interrupt, the model would stop time to run the code for the hub, causing simulations to take incredibly long to complete. Instead, the model doesn't run continuously, but only upon request arrivals at the ports. This allows the hub to simulate a continual scan of the ports without actually having to waste time checking ports when nothing has arrived.

The process model that **hub** executes is shown in Figure 4.7. As requests to send arrive at the hub's ports, they signal interrupts. The state **handle** keeps track of which port it is currently servicing. If the current port has issued a request, then the request is accepted, and an acknowledgment that the hub is ready to transfer one packet is sent back to the station which issued the request. The station receives the hub's acknowledgment and issues a packet. The hub then forwards the packet to the destination port with some delay to account for the amount of time between when the hub accepts the packet and when it can start transmission. After forwarding the packet, the hub is ready to service the next port in order.

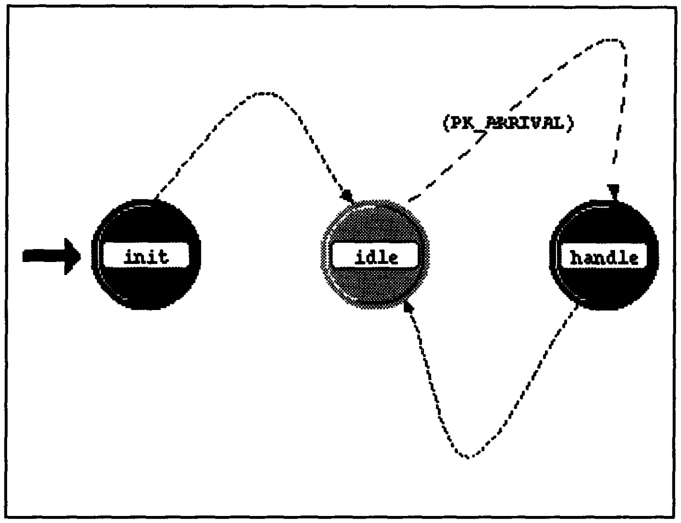

Figure 4.7: The 100VG-AnyLan hub process model.

Instead of forwarding the packet once all of it has arrived, the hub is cut-through and starts sending the packet before receiving all the bits that make up the packet. The hub starts transmission after getting enough bits to determine the destination address. In the structure of the ethernet frame Figure 4.8, the 48 destination address bits come after the 64 preamble bits. The hub must therefore wait until those 112 bits have come in before start-ing to transmit; so the modeled per packet delay is the time for 112 bits to travel on the 100Mbps wire, 1.12 μs.

Preamble	Destination	Source	Type	Data	CRC
64b	48b	48b	12b	46-1500B	24b

Figure 4.8: The Ethernet frame format.

4.4.3 The 100VG-AnyLan Station Model

The node model of a typical 100VG-AnyLan network station is shown in Figure 4.9. The processor-queue **vg_MAC** accepts data packets from a higher level source and encapsulates them in an ethernet frame before sending them to the network hub. The processor-

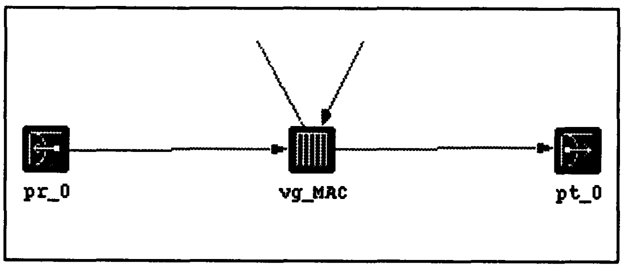

Figure 4.9: The 100VG-AnyLan Node Model

queue also accepts packets from the hub intended for the higher level sink (not shown) on the station, and disassembles them to forward only the data portion to the higher level sink. The receiver **pr_0** and transmitter **pt_0** are set to send and receive data at 100 Mbps.

The process model that **vg_MAC** executes is shown in Figure 4.10. It consists of five states: **init, idle, enqueue, send,** and **recv**. The **init** state is only run once at the start of the

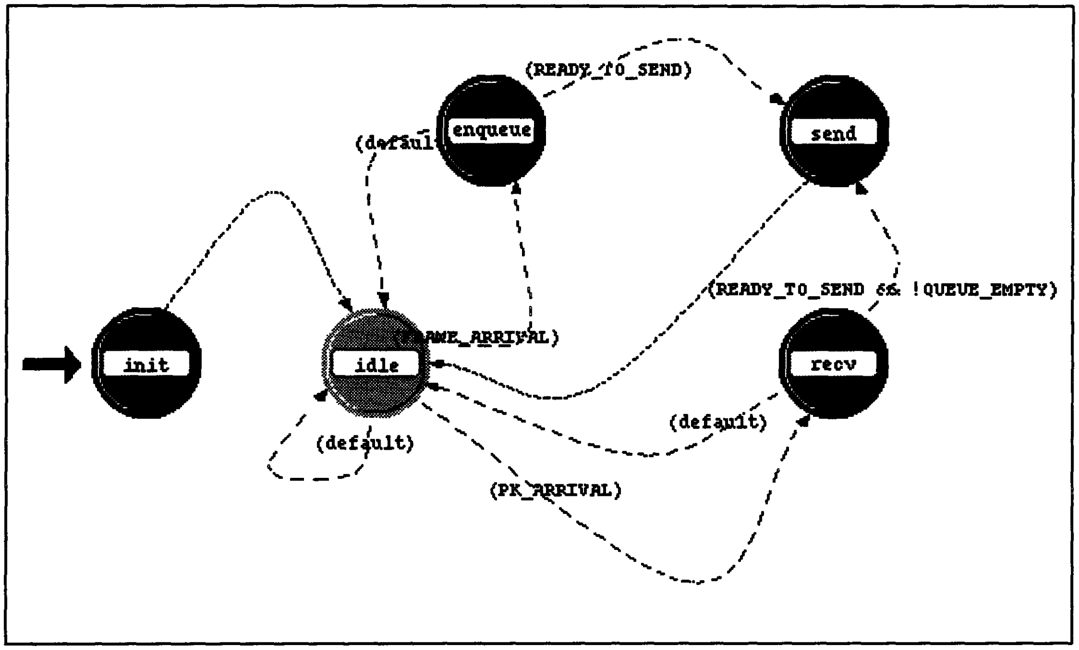

Figure 4.10: The **vg_MAC** process model

simulation and makes sure that the station knows its network port address. As data packets arrive from the higher level source, the **vg_MAC** processor transitions into the **enqueue** state where they are encapsulated into ethernet frames and stored in a queue until the processor is ready to forward them to the hub.

The **vg_MAC** process model keeps track of the state of relations with the hub using a flag called *ready_to_send.* If a station has already sent a request to the hub, then it must wait for an acknowledgment from the hub before forwarding a packet. If it is not waiting to send a packet, then it can go ahead and send a request to the hub to send a packet. If the station is ready to send, then the process model transitions to **send** where a request to send is issued to the hub and *ready_to_send* is set to FALSE.

When the station receives a packet from the hub, the **recv** state accepts it and determines whether the packet is a data packet intended for the station or an acknowledgment from the hub for a request made earlier by the station. If the packet is an acknowledgment then the hub is ready to accept a packet from the station, so **recv** dequeues a waiting frame and forwards it to the hub. **recv** then sets *ready_to_send* to TRUE so the process can submit another request to send. If, however, the packet is not an acknowledgment, then it is disassembled and the data portion is forwarded to the higher level sink.

4.4.4 The Lack of Priorities in the Model

Currently, the two separate priorities in 100VG-AnyLan are not being modeled, so all packets generated have the same priority. The priorities were not modeled because the only traffic on the network is the new image traffic from the ultrasound machines to the fileserver, and the stored image traffic from the fileserver to the reviewstation. The fileserver already gives preference to requests to write to the disk over requests to read, so in this situation, there is no need for priorities.

4.5 The SONOS Model

4.5.1 The Node Model

The model to simulate how the SONOS ultrasound machine would output data onto a network can be thought of in two separate parts: a processor which generates the ultrasound image data and a network interface which governs how data packets are put onto the network. For example, if the network medium were Ethernet, then the system could be thought of as a SONOS ultrasound machine connected to an Ethernet card, which in turn is connected to the network medium.

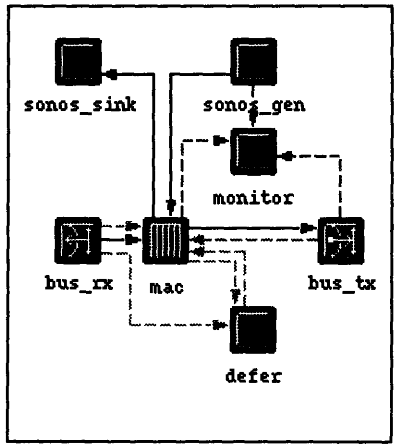

Figure 4.11: The SONOS node model (Fast Ethernet)

The node model of the SONOS ultrasound machine for Fast Ethernet can be seen above in Figure 4.11. It consists of four processors, a processor queue, a bus receiver, and a bus transmitter. The processor **sonos_gen** executes the *sonos* process model, and the processor **sonos_sink** is simply a data sink, destroying whatever packets are sent to it. Currently it is not expected that the SONOS ultrasound machine provide any sort of network service, but if an ultrasound machine were to act as a server, those service requests would be dealt with by the **sonos_sink** processor. The transmitting and receiving of packets is controlled by the **mac** processor queue which controls all aspects of medium access, such as handling the exponential backoff in the Ethernet CSMA/CD algorithm, or sending requests to the hub in the 100VG-AnyLan network. The processor **monitor** outputs interesting performance metrics about the ultrasound machine to a file for later analysis.

4.5.2 The Process Model

The **sonos_gen** process model, seen in Figure 4.12 as a state transition diagram, is composed of nine states: **init, gen, Dop, CF_Dop, CF, 2D, Gap, still,** and **idle**. The **init** state is only visited once for each SONOS ultrasound machine on the network at the start of the simulation, and sets up all the information which will be needed by the other states as time passes. **init** determines the network address of the individual SONOS machine, sets up an interface for communication with the medium access layer (MAC), and sets up the probability distributions for the image traffic generation to be used by the other states. The transition from **init** to **gen** is unconditional.

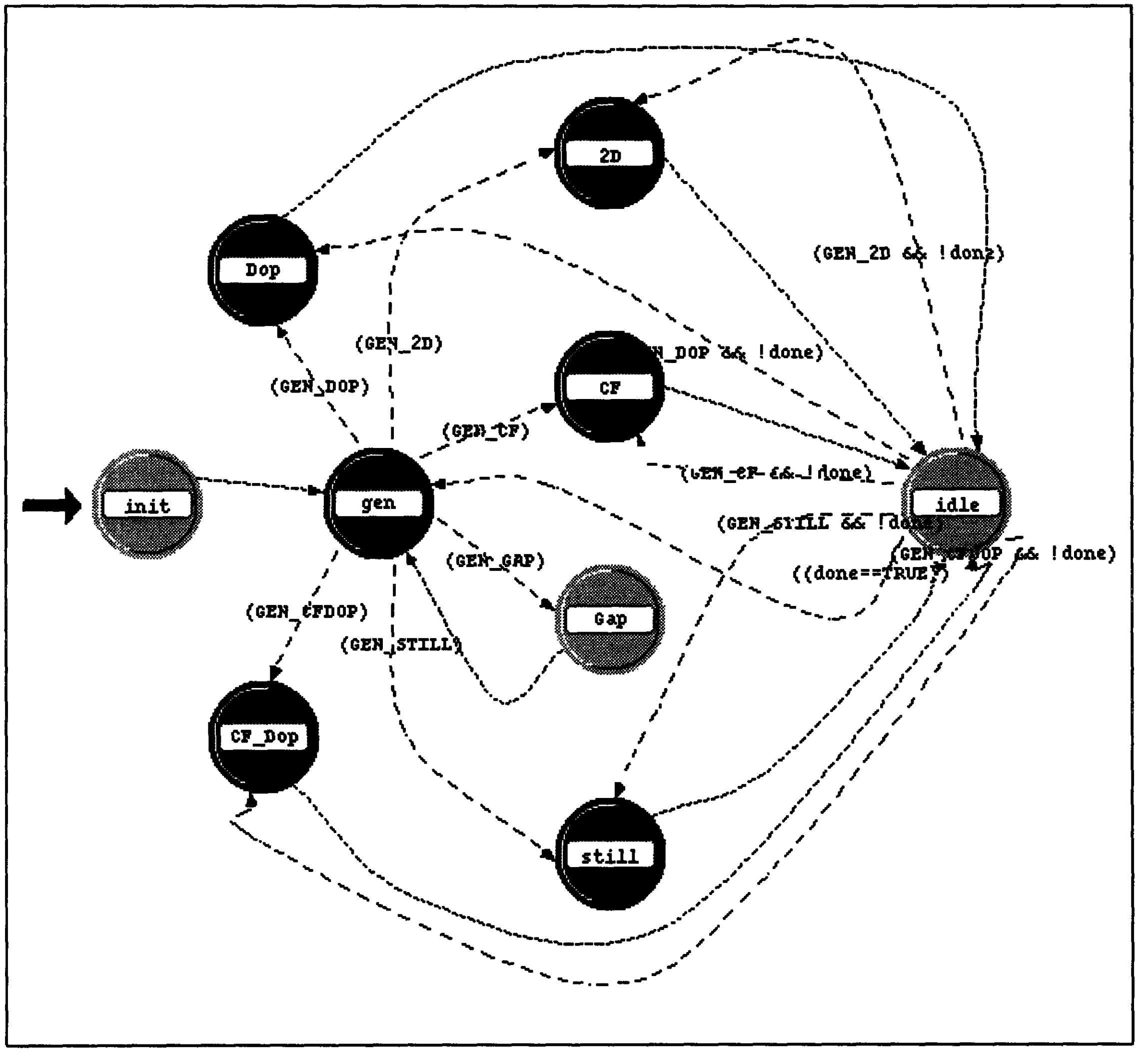

Figure 4.12: The SONOS process model state transition diagram

The span length and mode distributions which init loads into memory is that of the empirical data from the analysis of the ultrasound machine usage patterns discussed in Chapter 3:"The Workload Model". Since the usage patterns when taking adult and pediatric exams were found to be similar, only the adult workload is used in the simulation model.

The **gen** state determines which type of data should be generated, and for how long by using the probability distributions set up during the **init** state. Since the model simulates only the generation of the ultrasound image data, and not the data itself, the actual mode is irrelevant. What is important, however, is the frame rate at which that data is generated. Also of importance is whether the machine is not generating data at all, which happens during a gap. The transition to one of the six generation states (**2D, CF, Dop, CF_Dop, Gap, still**) is dependent upon the mode determination made by **gen**.

The generation states merely create and transmit the required number of packets to simulate the SONOS data rate depending on the mode. For a 512x512 greyscale ultrasound image, this gives about 2.1Mb per frame because each pixel takes up one byte. When creating 2D data, the SONOS ultrasound machines generate 30 fps thus pushing about 62.9 Mb of data onto the network per second. The color flow mode generates at only 15 fps; but since colorflow uses two bytes per pixel, it also dumps about 62.9Mbps onto the network. Better data compression can be achieved with colorflow data than with 2D data, however, so the ultrasound machines would actually generate less information when taking color flow data than when taking 2D data. Current data compression allows for 3x lossless compression for 2D, and 5x lossless compression for color flow.

Although Doppler and M-mode are distinct as far as the real information generated from a cardiologist's point of view, they are identical from the viewpoint of the SONOS process model. Both modes generate data such that a new screen appears every 2.4 seconds. In the process model, generating both M-mode and Doppler (2D) would cause a transition to **Dop**, whereas generating color flow Doppler would cause a transition to **CF_Dop**. The M-mode and Doppler modes generate about .87 Mbps, in contrast to the high data rate sustained by the ultrasound machines when taking 2D or color flow data.

During a gap, no data is generated. For a still shot, only one frame is generated, and then nothing for the duration of the view.

Although the Sonos ultrasound machines generate continuously (excepting the gap state) the process model must have some unforced states so that control can be passed back to the simulation kernel. Aside from **still**, the states that actually generate data are forced. The generating states know how many packets they must generate per second, and spread out the packet generation evenly over the course of a second, so that the overall data rate is correct. The amount of time to wait after sending a packet before sending the next is one second divided by the number of packets to send per second. Each generation state generates only one packet per activation, then transitions to the idle state so that the simulation kernel can attend to other duties. The generation states schedule self interrupts with the kernel so they can resume operation after the specific amount of time has passed.

4.5.3 Sonos Model Assumptions

Although the SONOS process model does not know whether it is being used on Fast Ethernet or 100VG-AnyLan, it does know that it is using ethernet style packets. The maximum size data segment allowed in an ethernet packet is 12,000 bits. The data size used by the model, however, is 10,240 bits, for reasons that will be explained in Section 4.6: "The File Server". Both the 100VG-AnyLan and Fast Ethernet medium access (MAC) layers encapsulate the packets from the Sonos process model into Ethernet frames, and then forward them on to the network by placing them in their transmitters' input streams. After leaving the medium access layer, the completed Ethernet packets placed onto the network are 10,436 bits in length.

Currently, there is no explicit buffer in the Sonos node model. The **sonos_gen** processor creates packets and then sends them to the MAC processor-queue. The packets wait in the queue until the MAC layer can shovel them off to the network. The buffering requirements of the ultrasound machine can be determined by watching the growth of the size of the MAC queue and the amount of packets waiting in the transmitter's input stream.

4.5.4 The SONOS Monitor[1]

In an effort to gauge certain performance metrics, the model uses processes to randomly sample the items of interest, and write them out as matlab program files for later analysis using matlab. To show the offered network load, the processor **monitor** randomly samples the bit throughput through the ultrasound machine's network transmitter. The number of bytes currently stored in the ultrasound machine's local buffer are determined from the number of data packets stored in the queue-processor **mac** (or **vg_mac** for 100VG) and the number of Ethernet frames waiting for transmission in the transmitter's input stream.

4.6 The File Server

A simplistic model of a file server was developed merely to control access to the hard disk. A typical hard disk, like the HP 97560, would not be sitting on the network by itself, because the IMACS network medium would most likely be 100VG-AnyLan or Fast Ethernet, and hard disks typically use the SCSI (Small Computer Serial Interface) to communicate with a network workstation acting as a file server. The file server in the echolab IMACS sits on the network and accepts information from the SONOS ultrasound machines to be digitally saved. The fileserver can be connected to a number of hard disks on a SCSI-2 bus, but this model assumes the use of only one large disk on the SCSI -2 bus.

The node model of the file server, shown in Figure 4.13 for 100VG-AnyLan, consists of the **monitor**[2] processor, two queue-processors, and two transceiver pairs. The queue-

---

1. The use monitors in general is discussed in Section 4.9: "Modelling Notes".
2. Although each node with a monitor process (fileserver, disk, reviewstation, and sonos) has a processor called **monitor**, the monitors are all different. They watch different items of interest and run different C code.

processor **vg_mac**, controls the access to the network by the queue-processor **fileserver** which both accepts packets from and sends packets to the network. The pair (**pr_0** and **pt_0**) of point-to-point transceivers is used to exchange ethernet packets with the 100VG-

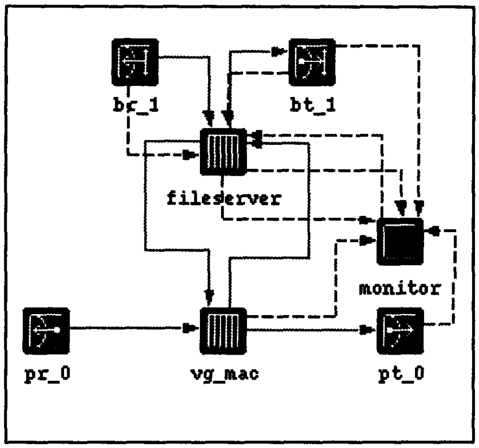

Figure 4.13: The fileserver node model (100VG-AnyLan)

AnyLan network. (In the case of a fileserver on Fast Ethernet, these transceivers would be bus transceivers instead of point-to-point.) The pair (**br_1** and **bt_1**) of bus transceivers is used to communicate with the disk over a SCSI-2 bus. The processor **fileserver** handles the transfer of ultrasound image data to the disk, as well as the transfer of the image data to a reviewstation.

4.6.1 The **fileserver** Process Model

The state transition diagram describing the fileserver process model is shown below in Figure 4.14. The diagram consists of five states: **init, idle, 2net, 2disk, and ops.**

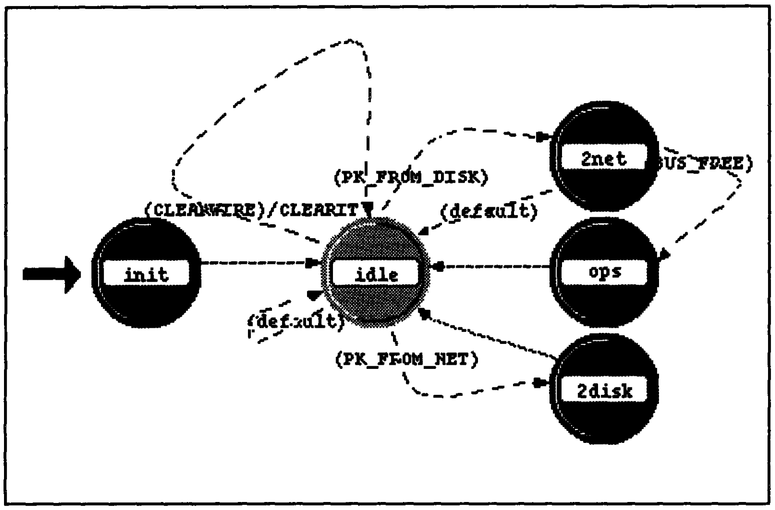

Figure 4.14: The **fileserver** process model

The init state initializes a number of variables used to keep track of the state of the fileserver. For example, the fileserver has to know whether it is currently servicing a request for an exam from a reviewstation, so it needs to keep track of the number of bits already sent to the reviewstation.

OPNET allows packets to be created in specific formats, so all the data packets created by stations on the network are created in a format called *imacsformat.* This allows all the stations on the network to indicate which type of station created the packet by setting the value of the field *initiator.* The value of *initiator* is one of three values corresponding to "ultrasound machine", "reviewstation", and "fileserver". The *imacsformat* packet is then encapsulated in an ethernet frame by the stations' medium access control processes. When an ethernet frame arrives at the fileserver from the network, the MAC processor extracts the data portion and forwards it to **fileserver.**

4.6.1.1 Fileserver to Disk (fs2disk)

Normally software running on the fileserver would be able to determine what file a workstation is requesting and where it is located, or to which file an ultrasound is trying to write. It would be impractical, however, for this model to attempt to simulate a network file system; so the state **fs2disk** instead determines whether the packet is image data or a request is made by looking at the *initiator* field in the packet format. Packets from the reviewstation are taken to be requests for exams, and packets from the ultrasound machines are assumed to be image data for writing to the disk.

If the packet were sent from a reviewstation, then **fs2disk** destroys the packet, and sets the state variable *servicing_request* to TRUE. Additional requests from other reviewstations are queued, until the fileserver has finished sending an exam's image data to the reviewstations. The size of an exam is assumed to be ~9,357 Mb, and was estimated from the average exam breakdown.[1] When fs2disk receives a request from the fileserver for an exam, it adds 9,357 Mb to the state variable *bits_to_get*. Since there is no meaning to the data sent back to the reviewstations, and only the amount of data sent back is important, the image data bits that the fileserver gets from the disk to send to the reviewstations are split uniformly between the reviewstations which have submitted requests for exams. This is because in the simulation, reviewstations cannot ask for an exam "by name", so they just ask for an exam. What is important is the number of bits received by a fileserver over time since it requested an exam.

If the packet were sent from an ultrasound machine, then it is assumed to be image data in need of writing to the disk. Because each packet coming into the fileserver from the network is in ethernet format, and too big for SCSI, **fs2disk** records the number of bits

1. The average exam takes 16:47 minutes, with 6:04 minutes of 2D, and 2:16 minutes of CF. Assuming both 2D and CF generate ~63 Mbps, and assuming data compression rates of 3 and 5 respectively, this leads to an estimation of a total exam size of 9357 Mb.

which should be sent to the disk in a state variable *bits_to_send*, and destroys the packet. The actual creation of packets for the SCSI bus is handled by the state **ops**.

Two performance measures of interest are the *network delay* and the *delay to disk*. The network delay is calculated by **fs2disk**, when a packet is received from the ultrasound machines, as the difference between the original packet creation time that the ultrasound stamps on it, and the current time. The delay to disk is the delay experienced by the image data between arriving at the fileserver and being written to the disk. When **fs2disk** receives an ethernet packet, it records the current time, which is then later used by the disk, upon receiving a block from the fileserver, to calculate the delay to disk.

4.6.1.2 Communication with the disk (**ops**)

The SCSI communication with the disk is handled by the state **ops**. **ops** recreates the data packets in *scsi_format* to send to the disk, and also tells the disk to begin a data transfer. The SCSI data transfers either 4KB or 512KB per block.

Whenever the bus goes free **ops** initiates a data transfer with the disk. Preference to disk writes is given over disk reads, so ops first checks the value of bits_to_send. If there are bits that need to be sent to the disk, **ops** creates a packet in *scsi_format*, sets the creation time of the destroyed *imacsformat* packet to which the newly created *scsi_format* packet corresponds, and sets the field *scsi_control* to indicate that a disk write is desired. The packet is then deposited on the SCSI bus. Otherwise, if no data is currently waiting to be written to the disk, as happens when the ultrasound machines experience gaps, **ops** checks to see if the fileserver is currently servicing an exam request from the reviewstation. If there are *bits_to_get,* and the fileserver is not currently waiting for the disk to send back data from a previous request, then ops creates a packet in *scsi_format* and sets *scsi_control* to indicate that a disk read is required.

Since the fileserver sends data to the disk whenever it has any, short blocks may be sent to the disk (especially in the case when the block size is large), until enough data has accumulated in the fileserver buffer that full blocks can be sent. An alternative policy would be to have the fileserver wait until it has enough data for a block to be sent. The advantage of this policy is that it would allow the fileserver to service the reviewstation since it would not be sending data to the disk. The disadvantage is that a system of time-outs would have to be used since it is possible that the data needed to fill a block may not soon arrive at the fileserver because the ultrasound machines enter a long gap, or stop generating data altogether at the end of an exam.

4.6.1.3 Fileserver To Network (**fs2net**)

The state **fs2net** controls how data is passed from the disk to the network through the fileserver. The amount of bits collected from the disk is stored in the state variable *bits_collected.* Overtime a *scsi_format* packet arrives from the disk to the fileserver over the SCSI bus, it is destroyed, and its size added to *bits_collected.* When enough bits have arrived to make an ethernet packet, **fs2net** creates a packet in *imacsformat.* The newly created packet is then sent randomly to one of the reviewstations which have submitted requests for exam data.

4.6.2 The Fileserver Monitor

The processor **monitor** measures the growth of the fileserver's buffer. The size of the buffer is determined by the size of the input queue to the fileserver's transmitters, the number of packets in the mac queue, and the state variables *bits_collected* and *bits_to_send.* **monitor** randomly samples this buffer size and writes out the total size in bytes to a matlab program file for later analysis and comparison with the other network statistics.

4.7 The Disk Model

In order to simplify the model, and not have to deal with issues such as SCSI bus arbitration among multiple disks, or splitting files among multiple disks, one large digital storage device was assumed. Actually the assumption is justified because currently very large RAIDS are being developed, capable of holding multiple ultrasound exams.

The node model of the disk drive, seen below in Figure 4.15, simply consists of two processors and a transceiver pair. The disk processor executes the **disk** process model, shown in Figure 4.16, which handles the ultrasound image information transfer to and from the fileserver over the SCSI bus. The pair (br_0 and bt_0) of bus transceivers handle

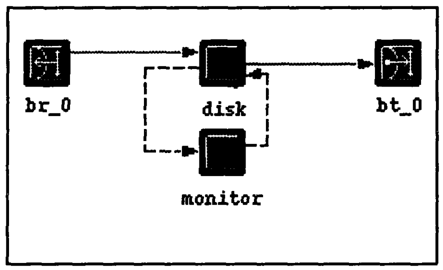

Figure 4.15: The disk node model.

the transfer of packets over the SCSI bus at the data rate of either 80 Mbps for normal SCSI-2 or 160 Mbps, since Fast Wide SCSI-2 is 20 MB/s. [Van Meter (28), §7.1.4]

The process model's state transition diagram, Figure 4.16, consists of four states: **init, idle, control, read, and write.** As SCSI packets arrive at the disk, the state transition dia-

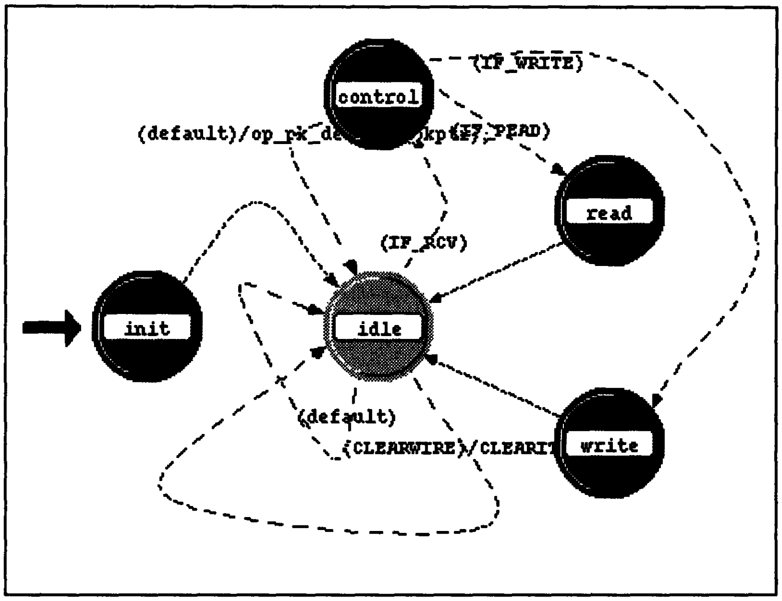

Figure 4.16: The disk process model.

gram moves to the control state where the state determines if the packet is ultrasound image data to be written to the disk, or if it is a request for image data. The **control** state looks at the field *SCSI_control* in the *scsi_format* packet which was set when the fileserver created the packets to send to the disk. Depending on the value of the field SCSI_control, the process model transitions to **read** or **write**. Before the actual transition takes place, however, the process model waits the appropriate amount of time to simulate the disk's seek time. The delay times used are the average random seek times for reads and writes, taken from the HP C2244/45/46/47 Disk Drive specification. [HP (10), 1-5]

4.8 The Review Workstation Model

The review workstation simulates the cardiology workstation on which a physician can log on in order to review the digitally stored ultrasound exams from the hard disks. Again, a simplistic model of the workstation is implemented. From a network performance perspective, the review workstation is nothing more than a black box on the network that sends requests to the file server for ultrasound exams.

4.8.1 The Node Model

The node model of the review workstation, Figure 4.17, is similar to the node model of the ultrasound machine. It is basically a sink, generator, and monitor connected to an interface to the network. The image data that comes in from the network is destroyed in **rvs_sink**. The processor **rvs_gen** generates requests to the fileserver for images. Currently, it is assumed that cardiologists will want to view ultrasound exams every ten minutes over a one to two hour period, twice a day. They typically read the morning exams after lunch, and the afternoon exams at the end of the day in concentrated sessions.

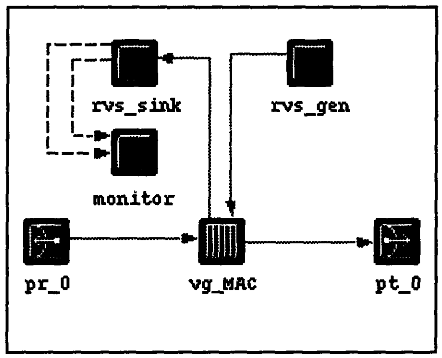

Figure 4.17: The reviewstation node model (100VG-Any-Lan)

4.8.2 The Process Models

The process model that **rvs_gen** executes is shown in Figure 4.18. On start-up, **init**

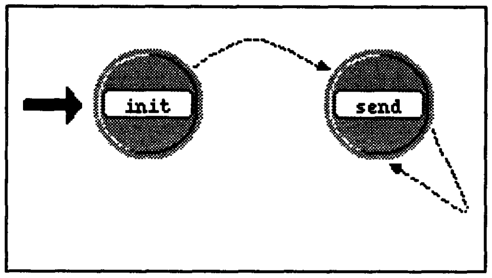

Figure 4.18: The reviewstation process model.

waits a random amount of time before transitioning to **send**. This is because different reviewstations may start up at different times, but once they are started, they may request to view ultrasound exams every ten minutes. Then the **send** state issues a request to the fileserver.

The process model for **rvs_sink** is shown below in Figure 4.19. The state discard

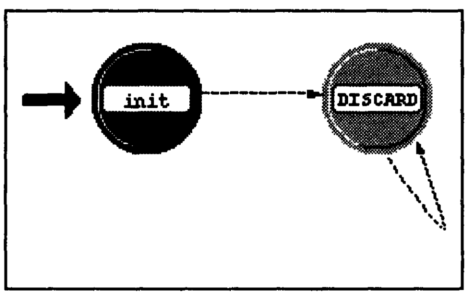

Figure 4.19: The **rvs_sink** process model.

keeps track of the number of bytes received so far from the fileserver and the delay that the packets experience. This information is then carried to the monitor processor over the statistic wires.

The monitor processor, **monitor**, watches values in the statistic wires concerning the end to end delay and the bits received by the reviewstation. This information is randomly sampled, and written out as matlab program files.

4.9 Modelling Notes

There are a number of general points about the simulation models which it is worth covering at this point. Some of the following notes are general assumptions that were not covered in the other subsections in this chapter, and some are interesting aspects of the program.

4.9.1 Random Numbers

Random numbers are extremely important in a simulation model, since the random numbers are used throughout the simulation for such things as implementing the actual issuance of requests or creation of packets. The IMACS simulation model described in this thesis uses random numbers to determine in which mode an ultrasound machine should be, and to determine the length of the span for that mode. The use of random numbers ensures that the distributions of the span lengths, and the modes are implemented fairly.

No random number generator, however, gives truly random numbers. All random number generators have a formula for generating a value based on some initial state information which changes as new values are generated. OPNET allows each run of a simulation model to use a specific value to seed the random number generator. This allows repeated runs using the same seed, to have the exact same network loading, and the same results.

Oddly enough, OPNET apparently does not use its own random number generator, but does the naive thing and uses the standard UNIX **random()** algorithm to generate the values for any calls to the OPNET random distribution functions. **random()** uses a nonlinear additive feedback random number generator employing a default table of size 31 long integers to return successive pseudorandom numbers in the range from 0 to $2^{31}-1$. Although **random()** is a very good random number generator, the problem is that the simulation model loses its ability to reproduce the same random numbers if the model setup is changed.

Since the OPNET distribution functions and random() all use the same state, any call to either changes that state, and therefore affects the values that can be expected. For example, two IMACS setups that differ only by the network medium used (one using Fast Ethernet and the other 100VG-AnyLan) would get different random numbers when run because all the objects in the model are using the same state to generate their random numbers, and the number of objects in both models differ since 100VG-AnyLan uses a central hub and pt-pt connections instead of a shared bus as in Fast Ethernet. This would make it difficult to compare identical runs on the two different networks.

The solution to the problem lies in making the model a multi-stream model, by using multiple random number generators. That is, not using the OPNET library distribution calls, or **random()** within the model of the ultrasound machine for the purpose of choosing a mode and span length when generating traffic. The ultrasound machine uses the same algorithm as **random()** to generate values, but uses a different state, so that the state is only changed by the ultrasound machines when creating packets to place on the network. This allows different IMACS simulation models with the same number of ultrasound machines to experience the exact same network loading.

4.9.2 Monitors

The various monitors discussed in the previous sections all write their output as matlab program files, instead of using the standard OPNET probes to capture such information as bit-thruput through the transmitters, delay times, and utilization. Handling the measurement of the various network metrics directly in the monitor processes allows each monitor to sample at a different rate. So for example, a receiver module in the fileserver node model does not cause a write out of measured information for every packet it receives, since the monitor actually does the write to the matlab file at the sample times.

Another benefit of using matlab program files to store the simulation results, is that the data can easily be analyzed and manipulated within matlab, which is substantially more powerful than the OPNET analysis tool.

The sample times of all the monitors are randomly determined using a uniform distribution defined as $U(0, \frac{1}{samplingrate})$ where the *samplingrate* may differ for each monitor process. The sampling times are randomly determined in an effort to avoid checking the metrics cyclically. For example, if the fileserver buffer size were only checked at periodic intervals, it would be more likely to miss important information, since every time it checks the buffer may be the same. Randomizing the sampling process greatly reduces this risk, although it is still possible that during a simulation run, a metric was never sampled when it was at its peak value. Future work would include adding state to the monitors so that local peaks or valleys in the measurements are recognized and written out as well.

4.9.2.1 Monitor Process Models

Although the code executed by each monitor in the simulation is different, the monitors share the same process model structure. That generic structure is covered in this subsection. Figure 4.20 shows the monitor process model for the fileserver.

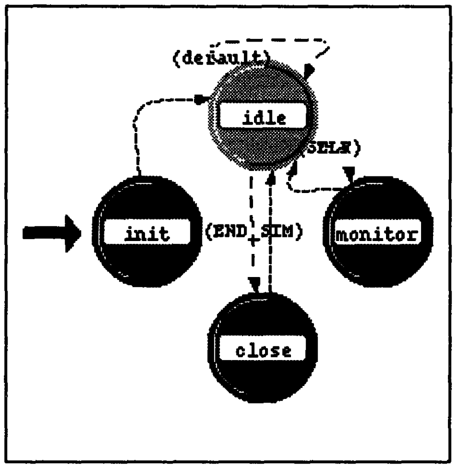

Figure 4.20: The **monitor** state transition diagram.

The **init** state opens the matlab files for writing by the **monitor** state, and sets the next process model interrupt using a random value from the normal distribution, so that the actual values written out to the matlab file are random samplings of the statistics of interest. The **monitor** state actually checks the values of its input statistic wires and writes them out to the matlab files. The **close** state terminates the matlab files with the proper commands to create and label the graphs of the statistics being measured.

4.10 Validation

Validation refers to ensuring that the assumptions used in developing the model are reasonable in that, if correctly implemented, the model would produce results close to what is observed from real systems. [Jain (12), 420] For the IMACS being simulated, however, no real system exists, so the simulation results cannot be compared to any real system measurements. An alternative method of validation involves the use of expert intuition to determine if the model assumptions and results are reasonable.

Several short simulation runs were conducted on the different network mediums, using different block sizes, and bus speeds, and a varying number of ultrasound machines. The short simulations used scripts telling the ultrasound machines exactly what data to generate and when. The graphs of the results of the simulations were analyzed to see if the reported buffer sizes and delay times were reasonable. Dr. Tennenhouse assisted greatly in looking into whether the results were consistent with what one would expect, given the various data rates of the network and the SCSI bus, the packet and block sizes involved, and the various delay times associated with the devices. This method was also useful as a debugging tool to verify that parts of the model were actually doing what they were intended to do.

4.10.1 A Sample Scripted Run

Test runs of the model were conducted using a script to tell the ultrasound machines what to generate instead of letting them determine what to generate based on the usage patterns determined from the empirical data. The use of such scripts allows the ultrasound machines to generate certain data rates at specific times. These tests were conducted to verify such things as whether collisions were causing backoff on the Fast Ethernet, whether the 100VG Hub was servicing all its requests without additional delay, and whether the fileserver was moving data to the disk at the proper rate.

The following graphs show the results from a test run of five ultrasound machines and one reviewstation on both 100VG and Fast Ethernet. Transfer to the disk was conducted over normal SCSI-2 using a small block size of 4KB. All five machines were started at slightly different times during the first second of simulation time, but each machine generates the same amount of data at the same rate. The load offered to the network (100VG and Fast Ethernet) by each ultrasound machine in this scripted run is shown in Figure 4.21. The scripted simulation run is 20 seconds long. The script tells each ultrasound machine to generate 2D data for the first three seconds, CF for the next three seconds, then generate nothing for three seconds. After the gap, each machine generates 2D for two seconds, then nothing for two seconds, then CF again for two seconds. For the last five sec-

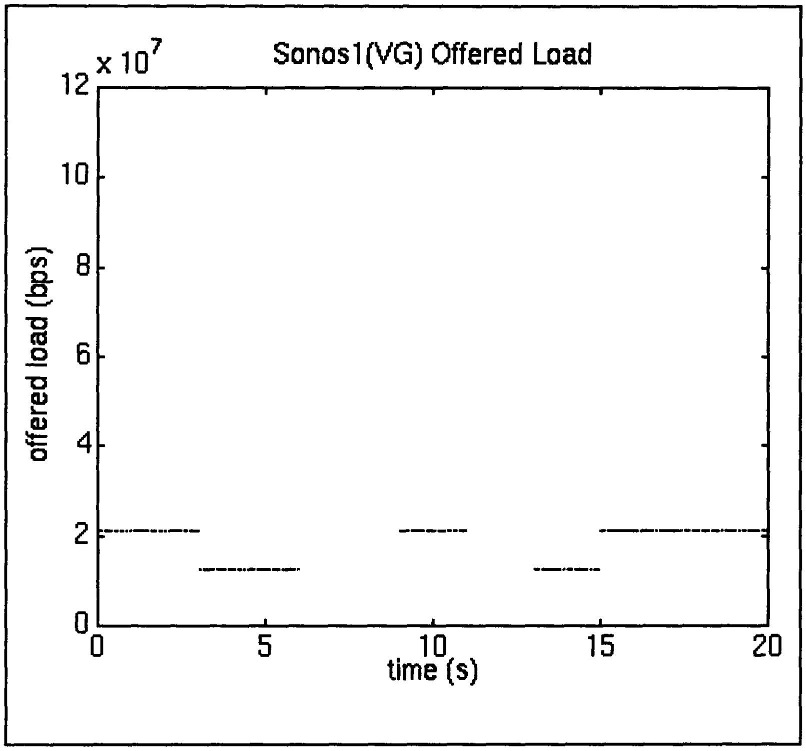

Figure 4.21: The offered load of each machine in the scripted run. The title "Sonos1(FE) Offered Load" means that this particular graph shows the offered load from Sonos #1 (of 5) on the Fast Ethernet network.

onds of the test simulation run, each machine generates 2D data.

The total load placed on the network by the ultrasound machines is shown in Figure 4.22. The graph shows that when all five of the ultrasound machines are generating in 2D mode, there is more than 100Mbps of offered load on the network. This is a substantial amount given that the data rate of the network medium is 100Mbps. Since the models of the ultrasound machines generate data at a constant periodic rate during a generation mode, packet collisions will always happen if all the machines start generating at the exact same time. Although test runs were conducted with all ultrasound machines having the same start time, this run allowed the machines to start up at random times during the first second. Therefore the chance of collisions is greatly reduced since each machine is phase shifted by the time it took to start generating.

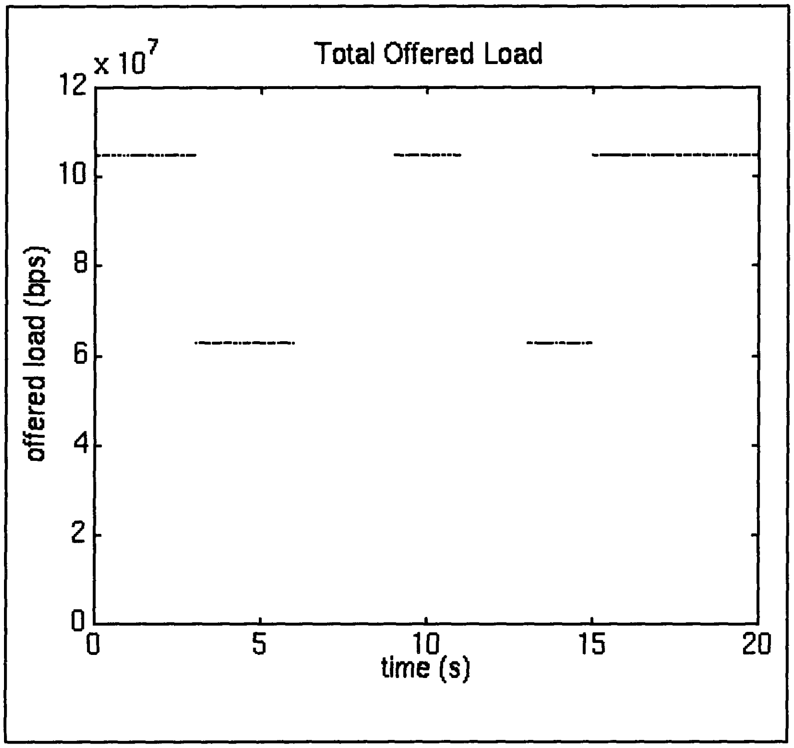

Figure 4.22: The total load offered to the IMACS network by the five ultrasounds machines in the scripted simulation run.

The two network mediums, Fast Ethernet and 100VG, have substantially different algorithms for conducting network traffic. It is expected that some packets travelling on the shared Fast Ethernet will experience more delay since in the event that the network is busy, a transmitting node backs off a random exponential amount of time. In the case of 100VG, however, packets are transferred one at a time The network delay experienced by the packets in the scripted simulation is shown in Figure 4.23. As the graphs show, each packet in the 100VG IMACS must wait no longer than it needs to, causing the delay to grow linearly if there are many packets to be sent. On the Fast Ethernet, however, due to the random backoff, the delay experienced by the packets does not grow linearly, and can be significantly greater than the 100VG network delay.

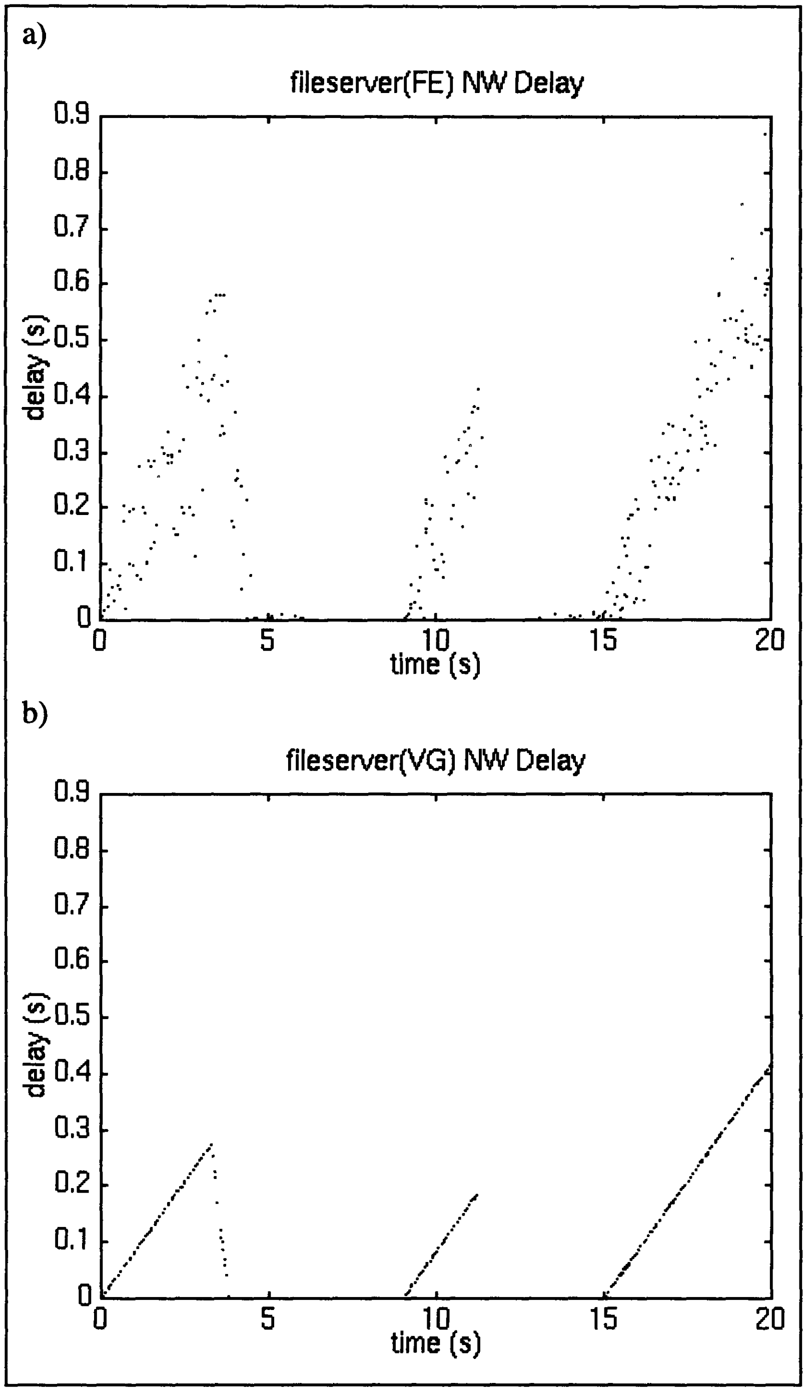

Figure 4.23: The network delay experienced by packets in the scripted simulation of 5 ultrasound machines on a) Fast Ethernet and b) 100VG network mediums.

The effect of having so many machines generate so much data at the same time is reflected in the growth of the buffers on the ultrasound machines and the growth of the fileserver buffer. The size over time of the ultrasound machine buffer is determined by the **monitor** processor in the **sonos** node model, Figure 4.11. The buffer size of one of the ultrasound machines in the scripted simulations on Fast Ethernet and 100VG is shown in Figure 4.24. The graphs show that the Sonos buffer size is larger on the Fast Ethernet IMACS because of the backoff that occurs when the machine is unable to dump its packet onto the network. On the 100VG IMACS, the buffer grows linearly when the machine is attempting to generate more than the network can absorb. In both graphs, the buffer size grows when all 5 machines are generating 2D, for example during the intervals: (0,3), (9,11), and (15,20). During gaps, or when the machines are generating color flow, the buffer size shrinks because the network is not being overloaded.

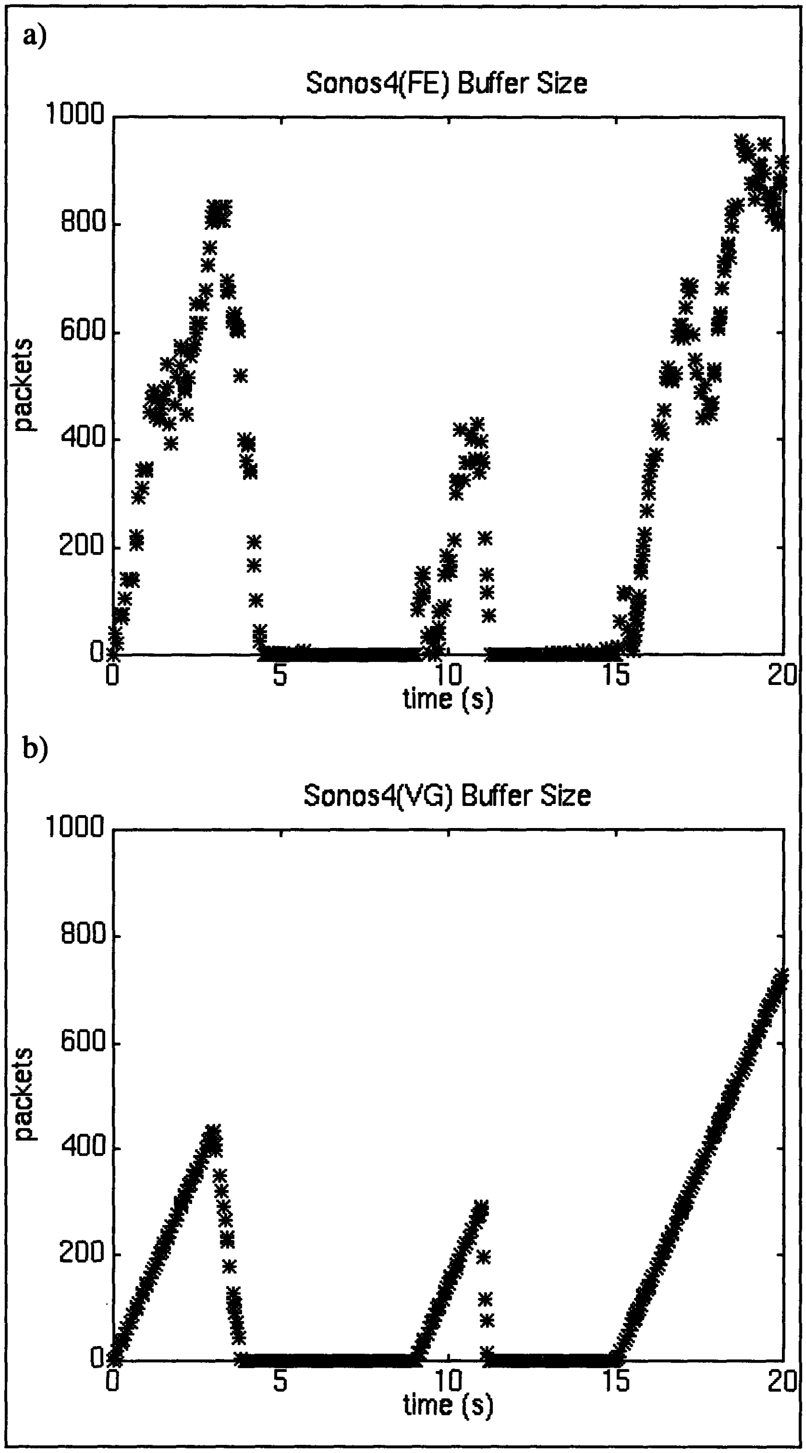

Figure 4.24: The growth of the buffer of Sonos4 on both a) Fast Ethernet and b) 100VG in the scripted simulation with 5 machines.

Another interesting difference between the buffer sizes on the 100VG network and the Fast Ethernet network is that on the 100VG, each machine's buffer grew identically since they were all following the same script and generating the same data. On the Fast Ethernet network, however, the buffer growth on the five machines were different. On the shared Fast Ethernet, when one machine is allowed to transmit its packet, the others must perform a random exponential backoff. On the 100VG network, in contrast, the machines are all serviced evenly, so it makes sense that the buffers of the Sonos on the 100VG network would be identical given that the machines are in the same data generation modes at the same time.

The growth of the fileserver buffer in the simulation run using a small block size of 4KB, shown in Figure 4.25, is reflected from the combination of how much data is coming in from the ultrasound machines, how much is leaving to the disk, and how much is being transferred from the disk to the reviewstation. The graphs show that the fileserver buffer grows greatly when all the machines are generating 2D. The buffer grows at a less steep

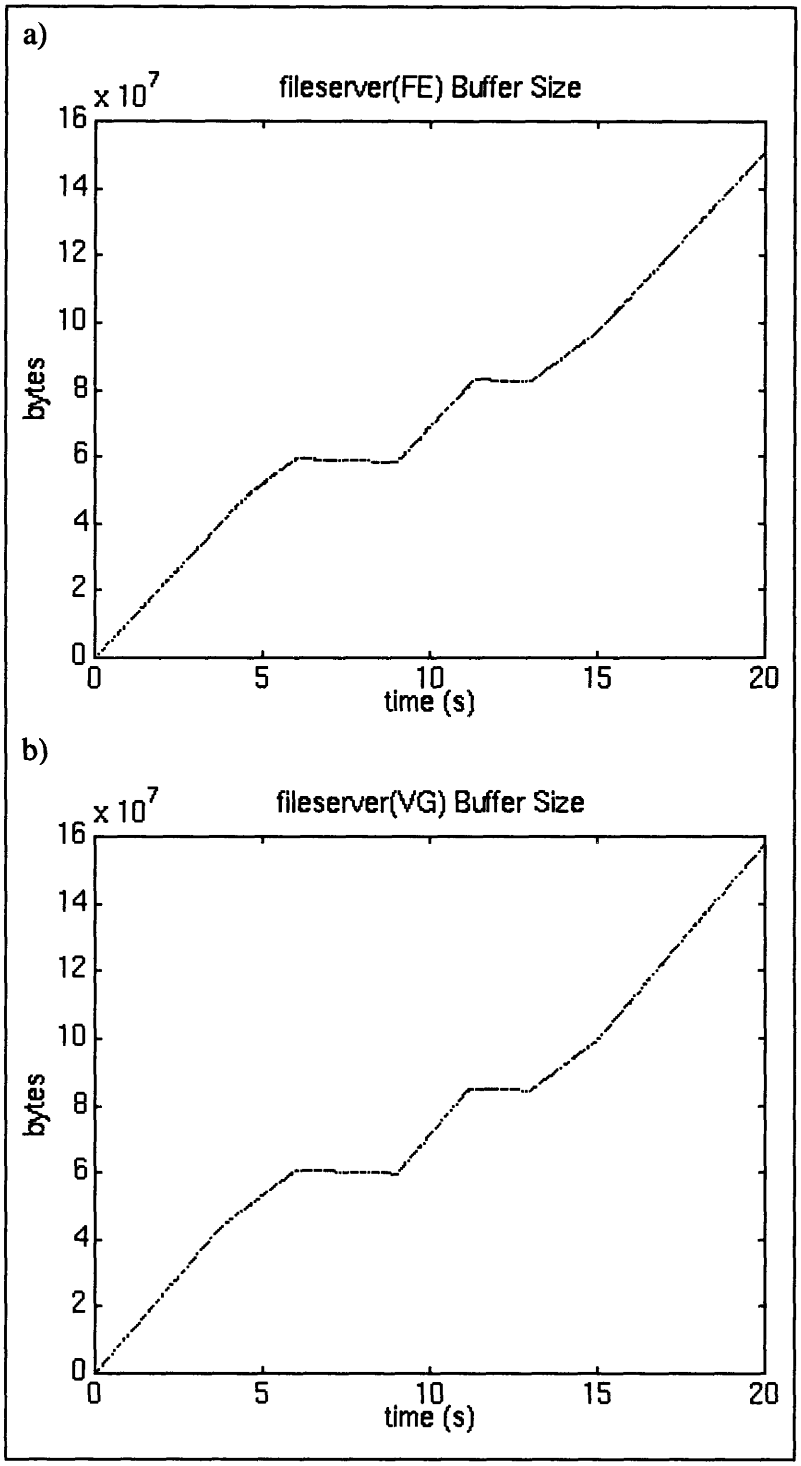

Figure 4.25: The fileserver buffer size on the two network mediums.

rate when they are all generating CF. The slow rate of transfer to the disk using a 4KB block, has a tremendous effect on the fileserver buffer. There is an 11ms seek/transfer time hit for every 4KB transfer to the disk. The result is that the fileserver buffer grows immense over the duration of the simulation, and since the fileserver always has data waiting to be sent to the disk, it never gets an opportunity during the 20 seconds of simulation time to service the reviewstation's request for an exam, even during the gaps when none of the ultrasound machines are generating, so during the simulation, the reviewstation gets absolutely no data from the fileserver.

The growth of the fileserver buffer from another run using a larger block size of 512KB for the SCSI transfer is shown in Figure 4.26. As the graphs show, because of the large block size, the fileserver is able to clear out its buffer to the disk during the gaps and the times when all the machines are generating CF. When using the large block size, the fileserver buffer is allowed to clear, giving the fileserver the chance to service the request from the reviewstation for exam data.

The use of the large block size also has a good effect, which is reflected in the graphs of the data received by the reviewstation over time, Figure 4.27. The reviewstation receives much more data back if the fileserver uses a large, rather than small, block for the transfer to and from the disk.

During the interval slightly before 10 seconds in the simulation, the reviewstation on the Fast Ethernet receives more data back from the fileserver than its 100VG counterpart does. This happens because at the end of the gap, both of the machines starts generating 2D again. The 100VG hub must service all three machines generating: the fileserver which is sending data to the reviewstation; and the two ultrasound machines which are trying to send to the fileserver. On the Fast Ethernet, however, the two ultrasound machines attempt to generate at their periodic rate, but collide with the fileserver's data and must backoff leaving the network free for the fileserver to send to the reviewstation. This effect of using a shared network is also discussed in Section 5.2.5: "IMACS configuration with 5 US machines, 160 Mbps SCSI-2, 512 KB block" in reference to Figure 5.28 on p.127.

The difference in the amounts received is minor; what is important is that the review-station received back data during the gaps when the large block size was used for the transfer to the disk, but did not receive any data at all when using the small block size.

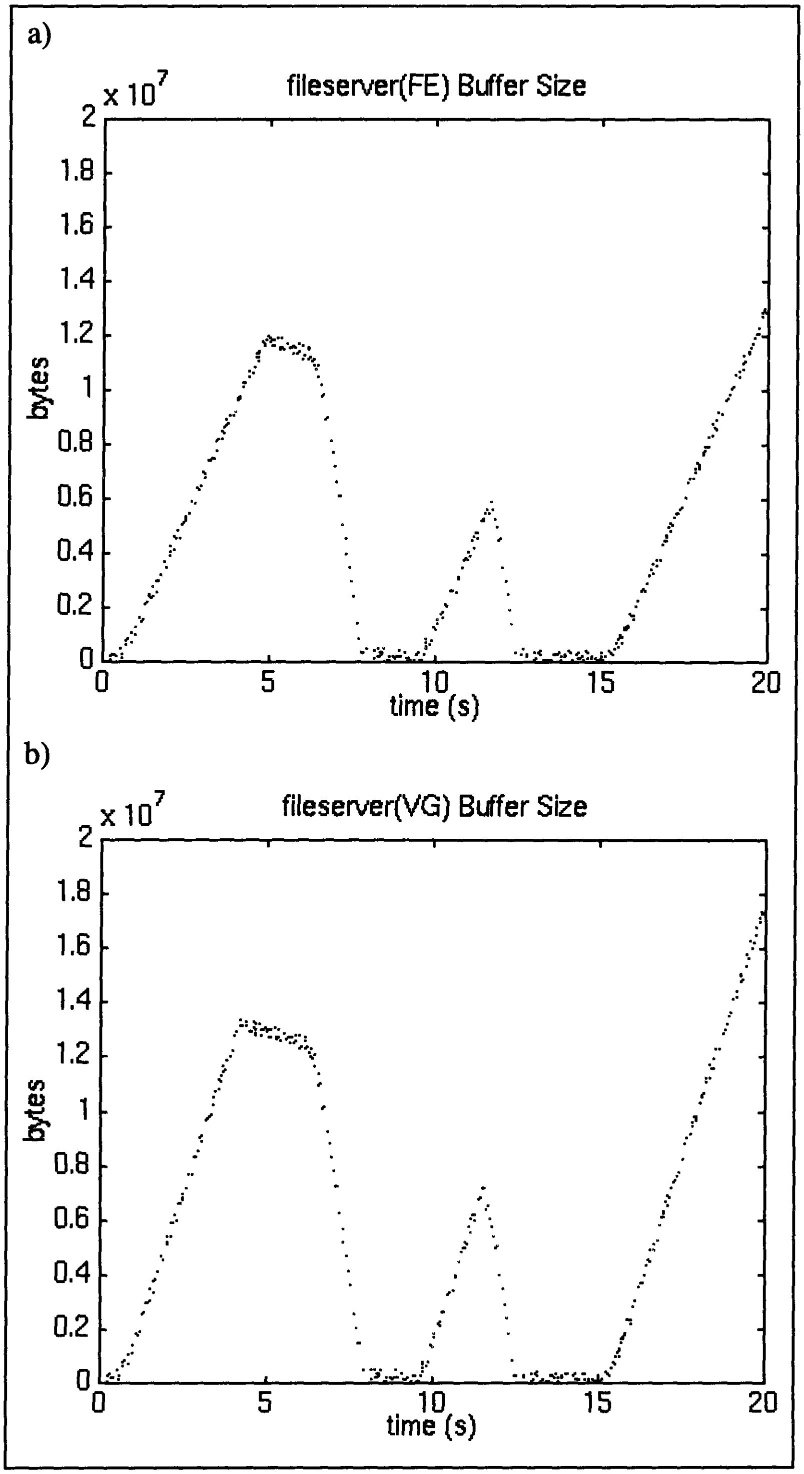

Figure 4.26: The growth of the fileserver buffer when using a large block size of 512KB for the SCSI-2 transfer in the scripted run with 5 ultrasound machines.

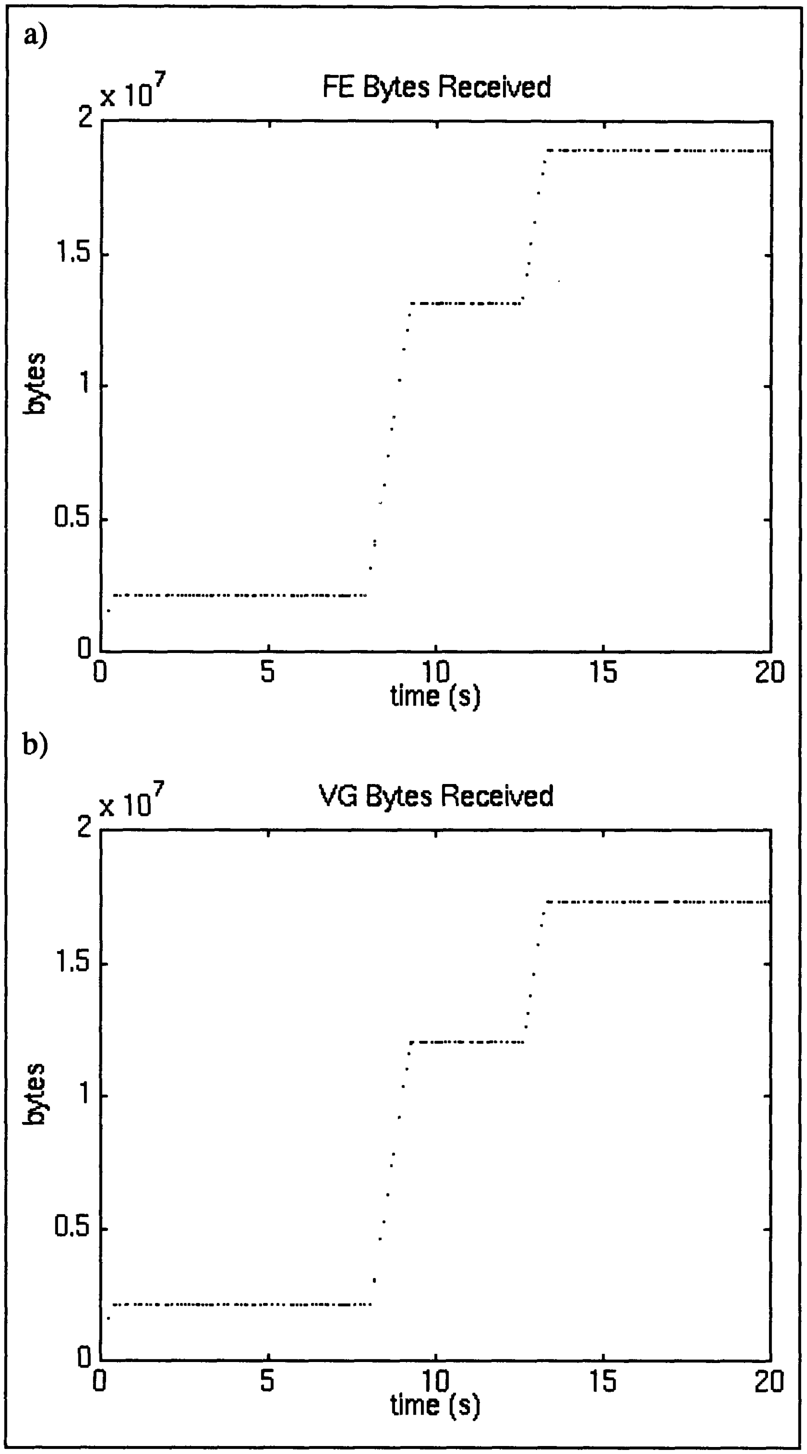

Figure 4.27: The amount of data received by the reviewstation over time when the fileserver is using a large block size for the transfer to the disk.

The network delay is measured at the fileserver as the time elapsed between a packet's creation at the ultrasound machines and the arrival at the fileserver. The delay times from

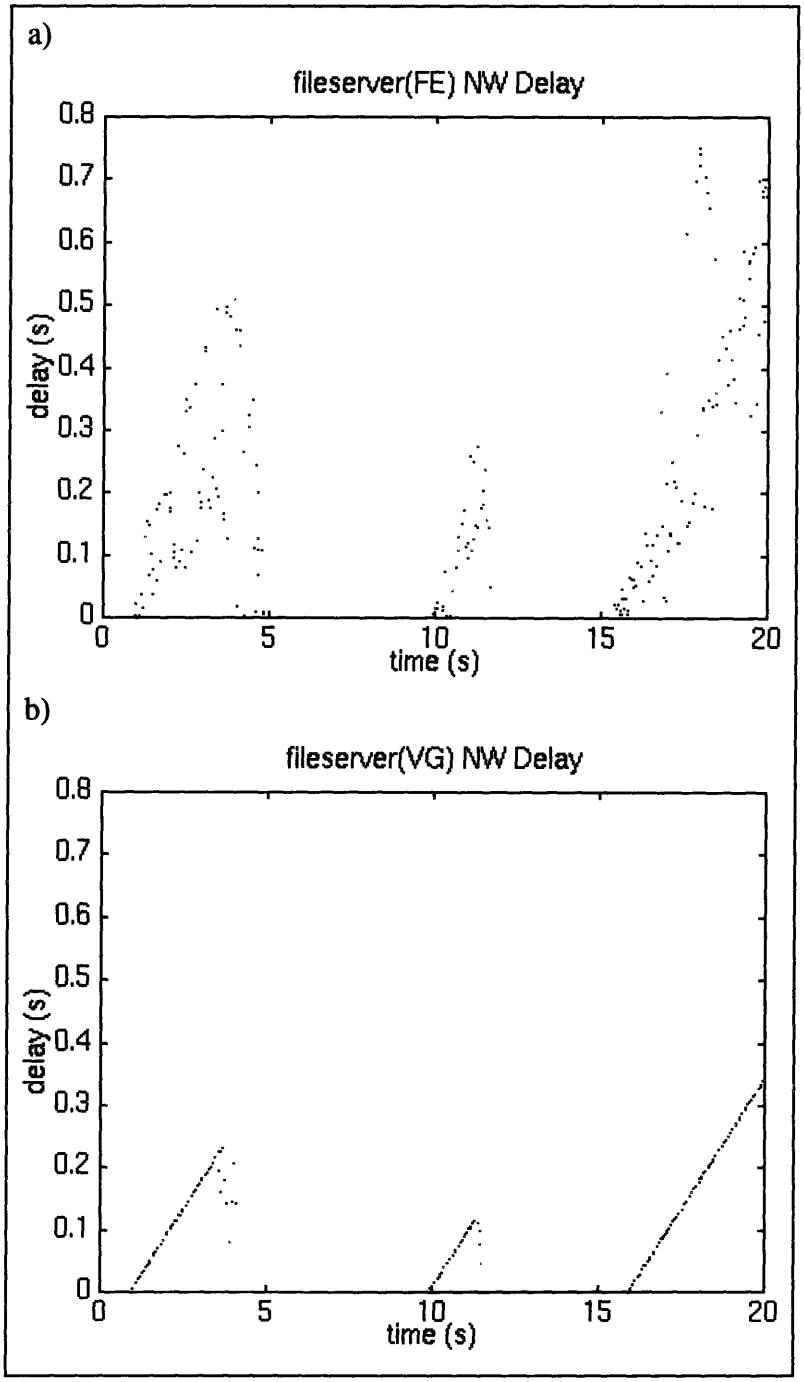

Figure 4.28: The network delay on a)Fast Ethernet and b)100VG

the run using the small block size were similar to the network delays from the run using the larger block size of 512KB since the block size primarily effects the transfer to the disk; although the delays were very slightly larger from the run using the large block size since the fileserver was able to receive data from the disk to send to the reviewstation. As one would expect, the delay on the Fast Ethernet is sometimes greater than the delay on the 100VG network. Moreover, the delay on the 100VG grows linearly, since the hub services the packets one at a time without any additional delay or backoff.

Once the data has arrived at the fileserver, the network medium is really of no consequence, but the speed of the transfer to the disk is. The delay that the data experiences between arriving at the fileserver, and arriving at the disk is shown in Figure 4.29. The graphs show the delay to disk experienced when using a small block size of 4KB and a large block size of 512KB for the normal SCSI-2 transfer to the disk.[1]

The delay to the disk on both the 100VG and Fast Ethernet simulations were similar; however, a big difference is seen due to the block size of the transfer to the disk. As can be seen in the graphs, when using a small block size, the delay really builds up. Essentially, when using a 4KB block size, almost 3 packets worth (assuming a packet is 12000b) can leave the fileserver to the disk every ~11ms. But if one ultrasound machine is generating in 2D mode, the fileserver can receive ~1750 packets in one second. This leads to quite a buildup especially if many machines are generating at the same time. The linearity of the delay to disk, using a 4KB block, seen in Figure 4.29 is due to the total offered load (Figure 4.22) placed on the network right at the beginning of the simulation; and can also be seen in the growth of the fileserver buffer during this time (Figure 4.24). The graphs show that when using a small block size, when information arrives at the fileserver it must wait an extremely long time before being written to the disk because there is so much data ahead of it in the fileserver buffer to be sent to the disk. In contrast, when using the larger

1. Please note the difference in the vertical scale between the two graphs. The delay to disk when using the 4KB block is graphed on a scale from 0 to 20 seconds, whereas for the figure corresponding to the larger 512KB block, the delay is no bigger than 1.8 seconds.

block size of 512KB, although the delay builds up as well, it is about an order of magnitude smaller, and the delay becomes smaller because the fileserver buffer clears during the gaps.

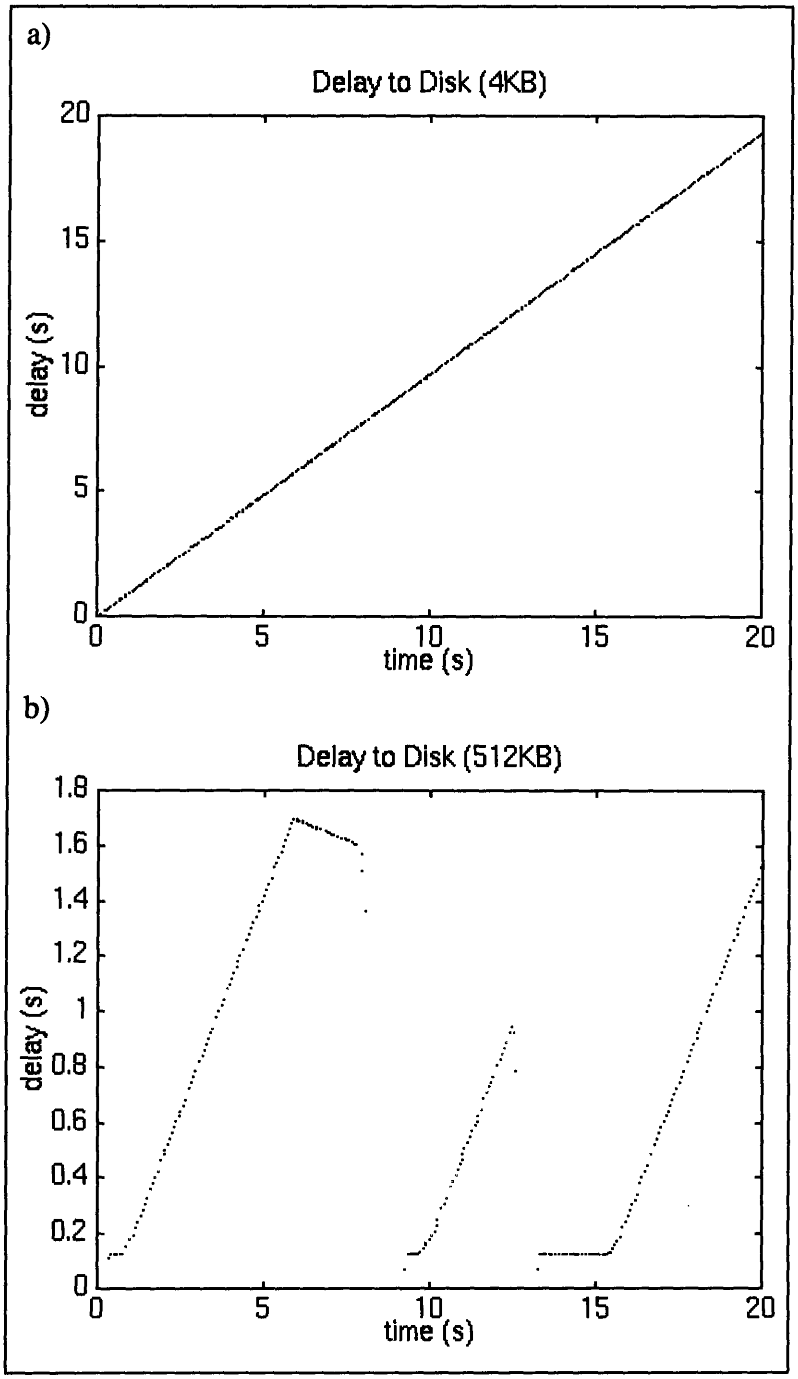

Figure 4.29: The delay to the disk using a block size of a)4KB and b)512KB.

4.11 Summary

The simulation model of the Image Management and Communication System is composed of an interconnected set of independent objects representing the network medium (Fast Ethernet or 100VG-AnyLan), the SONOS ultrasound machines, the reviewstations, the fileserver, and the disk. A specific configuration is captured by the choice of nodes at the network level mode. Each node on the network is described by its node level model, and each processor in a node is described by its process level model.

The network level model constrains how packets move between nodes. The node level model describes how information flows within the node itself. How that information is processed is described by the various process models executed by the node's processors and processor-queues. In addition, each node has a monitor process which contributes nothing to the actual operation of the node, but logs status information on various performance measures.

The simulation model of the whole IMACS system has been validated through a process of using scripted runs to force the ultrasound machines to generate specific modes at certain times instead of using the workload model discussed in Chapter 3:"The Workload Model". A number of performance measures consisting of network delay, delay to disk, fileserver buffer size, SONOS buffer size, and data recieved by the reviewstation were analyzed to see if the resultant values were credible. The validation process also helped in finding numerous flaws in the model, so that these problems could be debugged.

Chapter 5

Simulation Results

5.1 Experimental Design

A number of simulation executables with different network configurations (see Table 5.1) were developed and run on Sun SPARCstation-5 workstations under Solaris 2.3. Each configuration simulated required two runs. One simulation run involved a Fast Ethernet IMACS, and the other involved its 100VG-AnyLan counterpart. The simulation runs took from about 2 hours to over 8 hours of real time for just 5 minutes of simulated time. A total of 3 suites of simulation runs were conducted, although this chapter analyzes one such suite. The results from the other two runs were similar to what is discussed in this chapter, although a more detailed analysis of each run could be done in the future.

5.1.1 The Typical Case

Since cardiologists request exams about every 10 minutes, the time when the system must service both the generating ultrasound machines and the reviewstations is of importance. Each run simulates only 5 minutes in the life of an echolab in the typical case. Five minutes was chosen as the simulation duration because it is more than long enough for a reviewstation to have requested an exam, and received it. Ideally, if the network were totally free, and assuming an exam size of ~9357 Mb, it should take about 1.5 minutes for the review workstation to finish receiving the whole exam from the fileserver. Of course, the first bytes of the exam would be sent back almost immediately, and the reviewstation could display the exam as it gets them. Another consideration in limiting the simulation duration is the amount of real time it takes to run a simulation coupled with the limited computer resources available.

The worst case occurs when all the ultrasound machines are generating 2D data without gaps, and all the reviewstations are requesting exams. Basically, the worst case occurs when all the devices connected to the fileserver over the network are requesting service continually. In regards to coming to some conclusion about what kind of performance an echocardiography lab can expect from its IMACS system, the typical case is more important than the worst case for a number of reasons:

- From the analysis of the usage patterns, it is obvious that gap time is a significant component of the workload. Since the worst case workload contains no gaps, this fact makes the occurrence of the worst case extremely remote.
- The typical case is based on the *current* usage patterns of the ultrasound machines. In a digital echolab, sonographers will eventually learn that they do not need to store as much data as they currently do, because of the ability to run digital loops of a given number of heart beats. Therefore the "typical" case is in actuality already an extremely pessimistic view of the storage requirements.

5.1.2 The Simulation Executables

The simulation runs were conducted for each of the configurations shown in Table 5.1. Each entry in the table corresponds to a pair of runs, one on Fast Ethernet and the other on 100VG. The number of reviewstations was held constant at one, to focus on how the number of ultrasound machines, choice of SCSI bus, and block size affect the system. In each simulation, the reviewstation generates a request for an exam right at the start of the 5 minutes of simulation time. The file server attempts to service this request while giving priority to the ultrasound machines trying to save their data to the disk.

Although many of the configurations are not necessarily indicative of what an actual echocardiography lab may have, it is hoped that by obtaining performance data for a vari-

Number of US machines	SCSI-2 bus speed	Block Size
2	80 Mbps	4 KB
2	80 Mbps	512 KB
5	80 Mbps	4 KB
5	80 Mbps	512 KB
5	160 Mbps	512 KB
7	160 Mbps	512 KB

Table 5.1: The different network configurations simulated. **US** is the number of ultrasound machines on the network. The number of reviewstations was held constant at 1. The bus speed is either 80 Mbps for normal SCSI-2 or 160 Mbps for fast-wide.

ety of configurations, some light may be shed on how performance expectations change based on the configuration of the Image Management and Communication System.

5.2 Results

The following graphs show the results from two runs of the simulation for each of the configurations in Table 5.1. One run was of the configuration on the shared Fast Ethernet network medium, and the other on the switched 100VG network medium. All of the simulation runs used the same random number seed so that simulations using a configuration with the same number of ultrasound machines would experience the same offered load.

5.2.1 IMACS configuration with 2 US machines, 80 Mbps SCSI, 4 KB block size.

The configurations with only two ultrasound machines is underloaded because even if both machines are generating 2D data, that is only about 42 Mbps being dumped onto the network, and the network can easily absorb that; but it is still useful to look at it before moving on to the configurations with a higher number of ultrasound machines to get an idea of where system bottlenecks may be even with the underloading.

Each ultrasound machine on the network simulated the real usage of the Sonos by sonographers, using the distributions of the modes and span lengths discussed in Chapter 3:"The Workload Model". Both sets of runs of the simulation with two ultrasound machines use the same random number seed so that the ultrasound machines in the simulations generate the same data rates at the same times, and therefore both the 100VG and Fast Ethernet mediums experience the exact same network loading from the ultrasound machines. The offered load from one ultrasound machine, and the total load from both of the ultrasound machines on the system are shown, respectively, in Figure 5.1 and Figure 5.2.

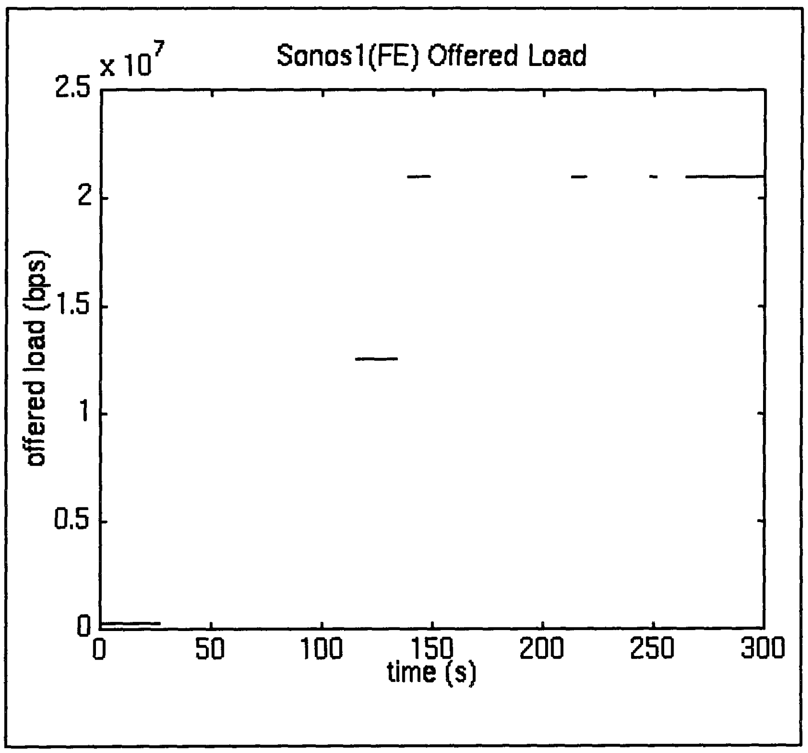

Figure 5.1: The offered load from Sonos1 in the simulations with two ultrasound machines.

As the total load on the system shows, there is very little time when both machines are generating 2D, although there are some large gaps.

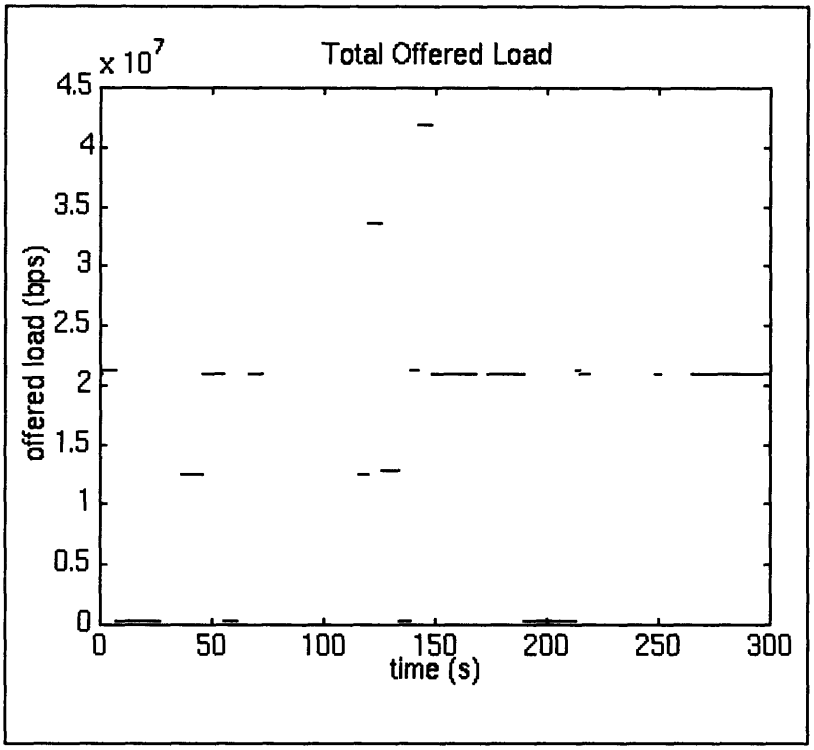

Figure 5.2: The total offered load from both ultrasound machines.

Because the network is not heavily loaded, and because the ultrasound machines start up at different times, but generate data periodically, if there is not a collision on the first packet, there probably will not be any collisions for the life of the simulation. The buffer size of one of the ultrasound machines on both 100VG and Fast Ethernet is shown in Figure 5.3. The graph of the buffer on the Fast Ethernet shows that a packet was never stored, which means that whenever this machine wanted to send a packet, the network was free. In the 100VG case, a packet is always stored momentarily while the station requests the hub for service, which explains why the buffer on the ultrasound machine on the 100VG network sees one packet in the buffer at times. The graphs of the buffer sizes of the ultrasound machines will be more interesting when more machines are contending for the network.

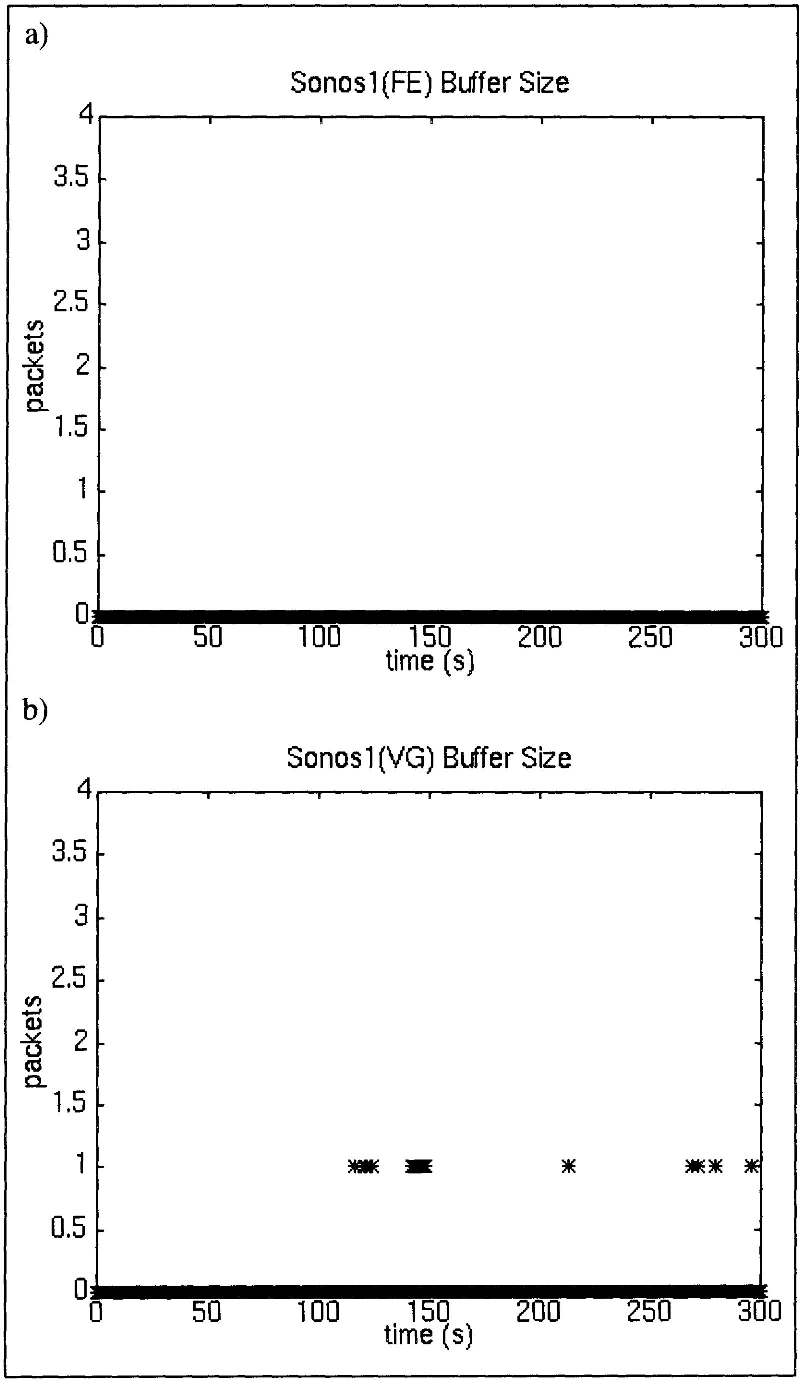

Figure 5.3: The buffer size graphs of one ultrasound machine from the simulation with 2 ultrasound machines, 80 Mbps SCSI-2, and a 4KB block.

The graphs of the delay between when packets were created, and received at the fileserver are shown in Figure 5.4. As the network delay graphs show, the packets travel-

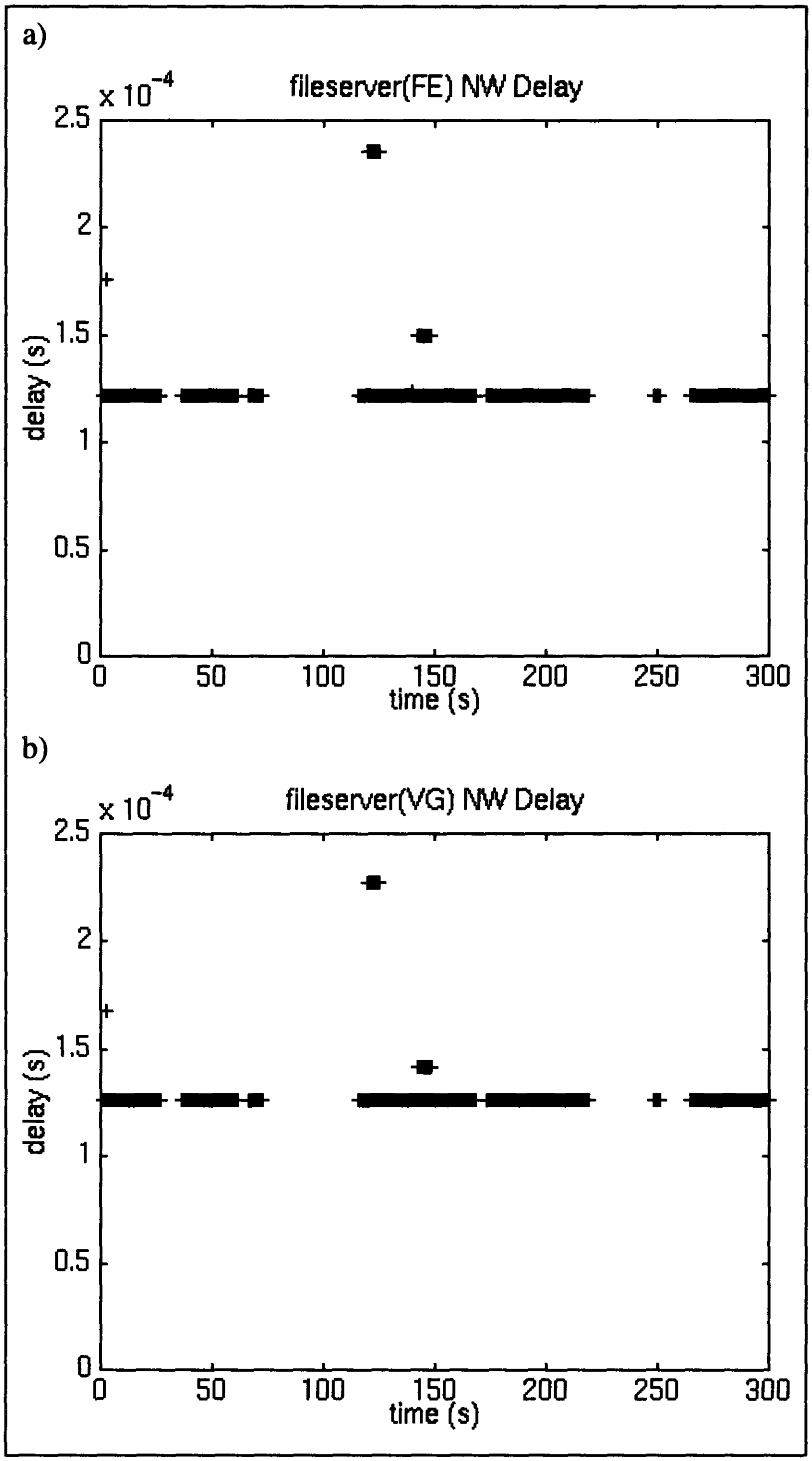

Figure 5.4: The network delay from the simulation with 2 US machines, 80 Mbps SCSI-2, and a 4KB block size.

ling on both networks experience the same network delay because there are very few collisions (the collisions appear to have happened when one machine was generating Doppler and the other CF). The network delay will also be much different between the two networks once more stations are pushing data onto the network.

The growth of the fileserver buffer for this run using an 80 Mbps transfer to the disk and a block size of 4KB is shown in Figure 5.5. Due to the small block size, the fileserver is unable to get much data to the disk at all. Even during long gaps, the fileserver is not able to clear its buffer. Clearly, in order to keep memory requirements reasonable on the fileserver, it is important that the fileserver be able to send the data intended for the disk to the disk. In this case, the fileserver ends up having to store most of that data over the course of the simulation, making it a huge bottleneck in the system.

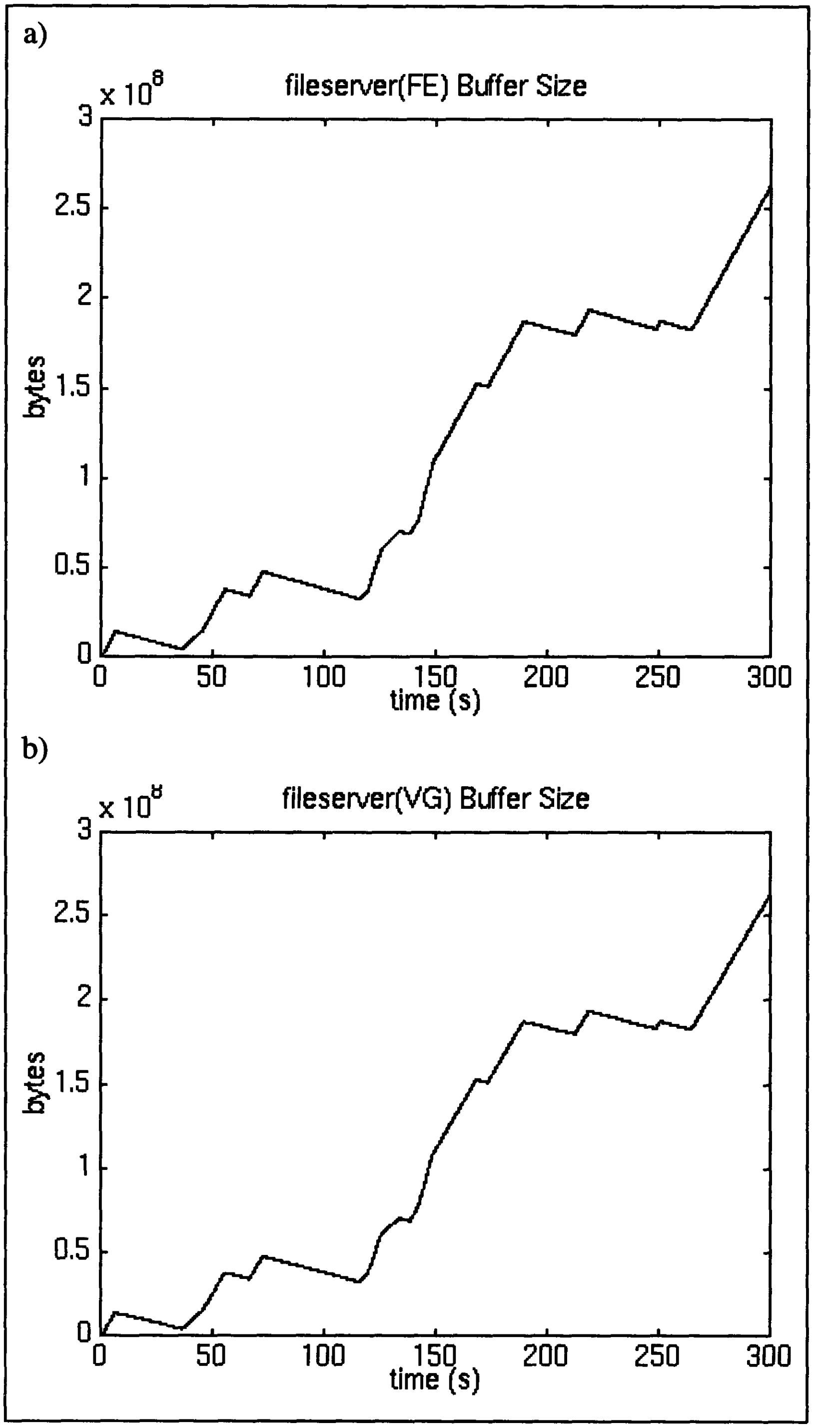

Figure 5.5: The buffer size of the fileserver for the simulation runs involving 2US machines, 80 Mbps SCSI-2, and a 4KB block size.

The delay that the information experiences while waiting in the fileserver before being written to the disk is shown in Figure 5.6. The reason for the buildup in the fileserver

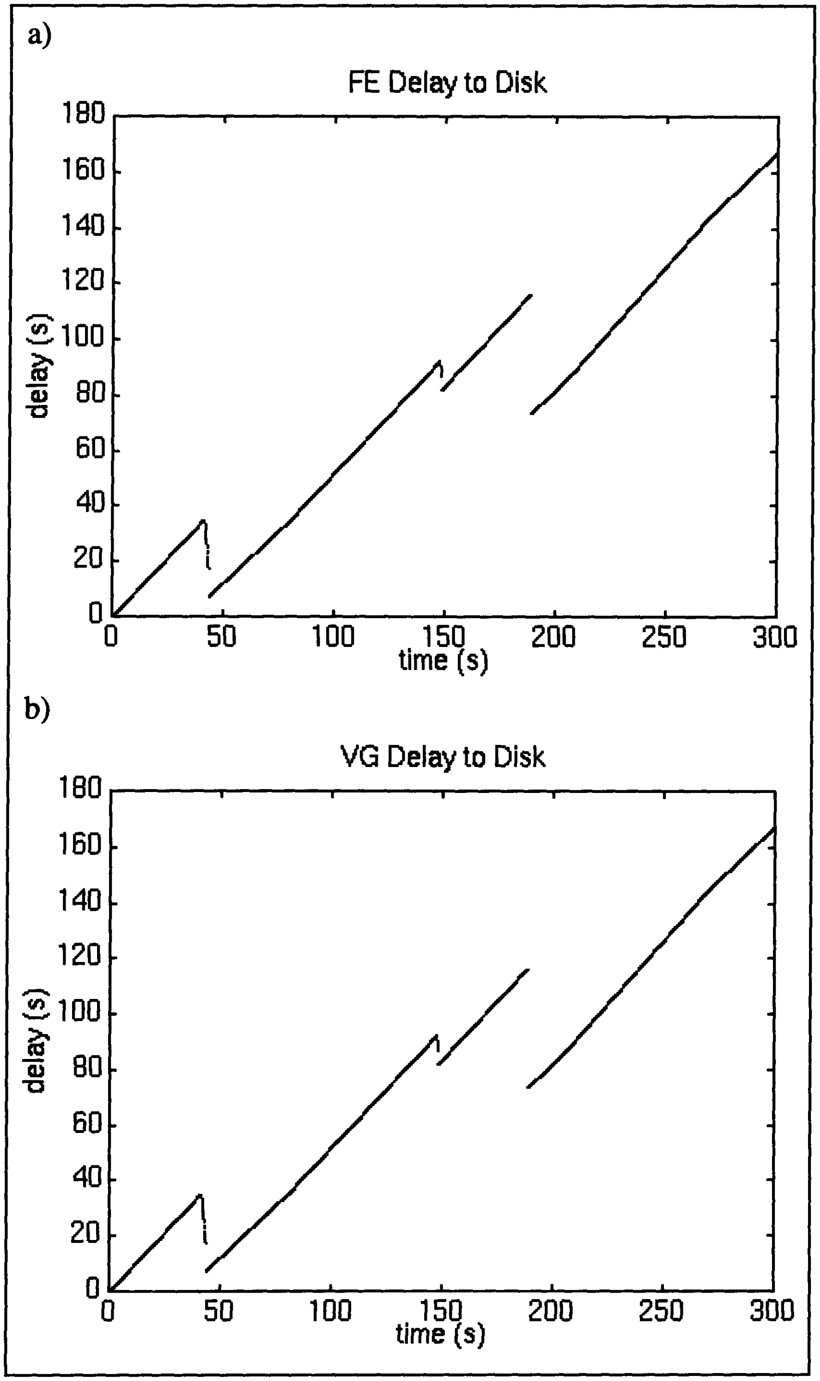

Figure 5.6: The delay to disk for the simulation with 2 US machines, 80 Mbps SCSI-2, and a 4 KB block.

buffer can be seen from the delay that the information experiences while waiting to be written to the disk. Data comes in to the fileserver at a much faster rate than it can be written out since each write to the disk incurs the hit of the disk seek time, and data is being written in such small blocks. Thus the time between when a packet arrives at the fileserver, and when it can be written to the disk grows, since each packet must wait before all of the ones that have arrived before it get written to the disk. Obviously, in order to get the data written to the disk in a more timely manner, a much larger block size must be used. Using a larger block size such as the 512KB block will also greatly alleviate the burden on the fileserver buffer.

The amount of data received by the reviewstation on both 100VG and Fast Ethernet is shown in Figure 5.7. The use of the small block also has a bad effect on the reviewstation.

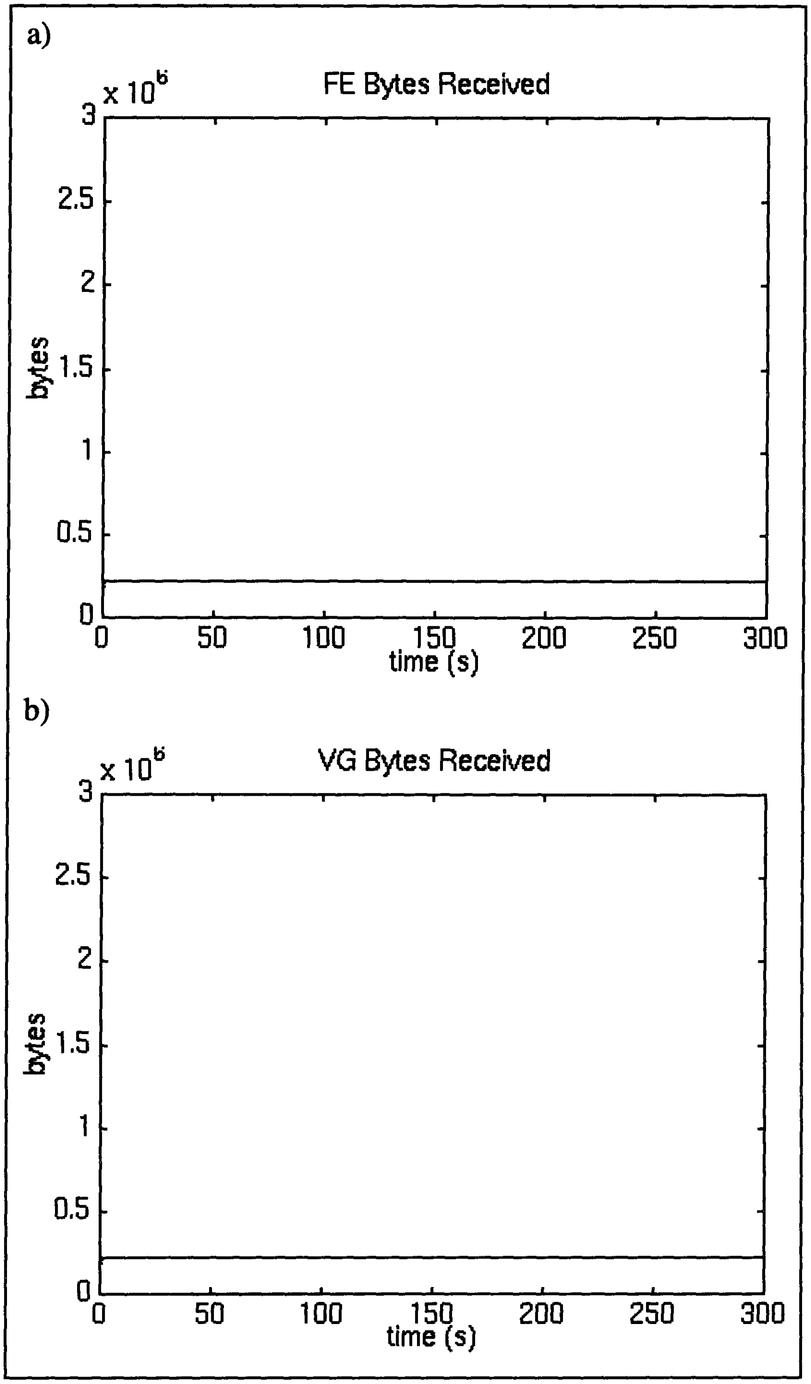

Figure 5.7: The bytes received by the reviewstation for the simulation involving 2 US machines, 80 Mbps SCSI-2, and a block size of 4KB.

Since the fileserver is never able to clear its buffer, it is never able to service the reviewstation's request for exam data. Aside from receiving some data right at the beginning of the simulation before the ultrasound machines start generating data, the reviewstation receives nothing over the course of the simulation. With a larger block size, the fileserver will be able to clear its buffer and service the reviewstation.

5.2.2 IMACS configuration with 2 US machines, 80 Mbps SCSI-2, 512 KB block.

This run is similar to the previous simulation run, with the only difference being the size of the block used to transfer data between the fileserver and the disk. In this pair of simulations, a large block of 512KB was used.

The offered load from the ultrasound machines was exactly the same as it was in the last simulation, so those graphs will not be reshown here. The buffer sizes of the Sonos in this simulation, however, were not the same as the last because in this run, the fileserver was able to send data to the reviewstation, adding load to the network. The graphs of the buffer sizes for one ultrasound machine on both 100VG and Fast Ethernet are shown in Figure 5.8. The added load from the fileserver does have a greater effect on the buffer on the Fast Ethernet ultrasound machine as can be seen in the graph where the buffer size hits ~15 packets. The buffer on the ultrasound machine on the 100VG network however, never sees more than one packet in its buffer because it never has to backoff unlike its Fast Ethernet counterpart.

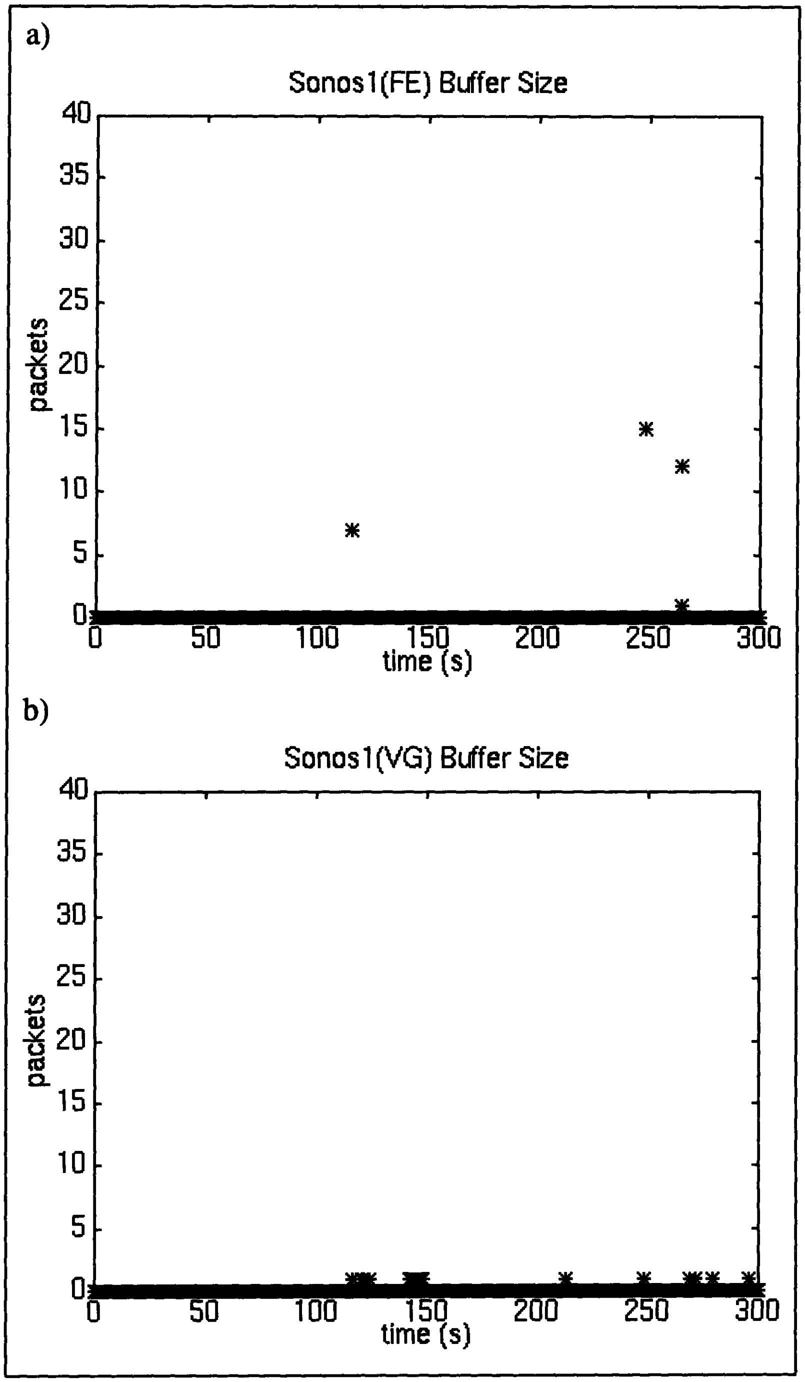

Figure 5.8: The growth of the buffer on one ultrasound machine on a)Fast Ethernet and b)100VG for the simulation with 2 US machines, 80 Mbps SCSI-2, and a 512 KB block.

The delay that the packets experience while traveling on the network is also different from the previous simulation. These graphs are shown in Figure 5.9. Both graphs are

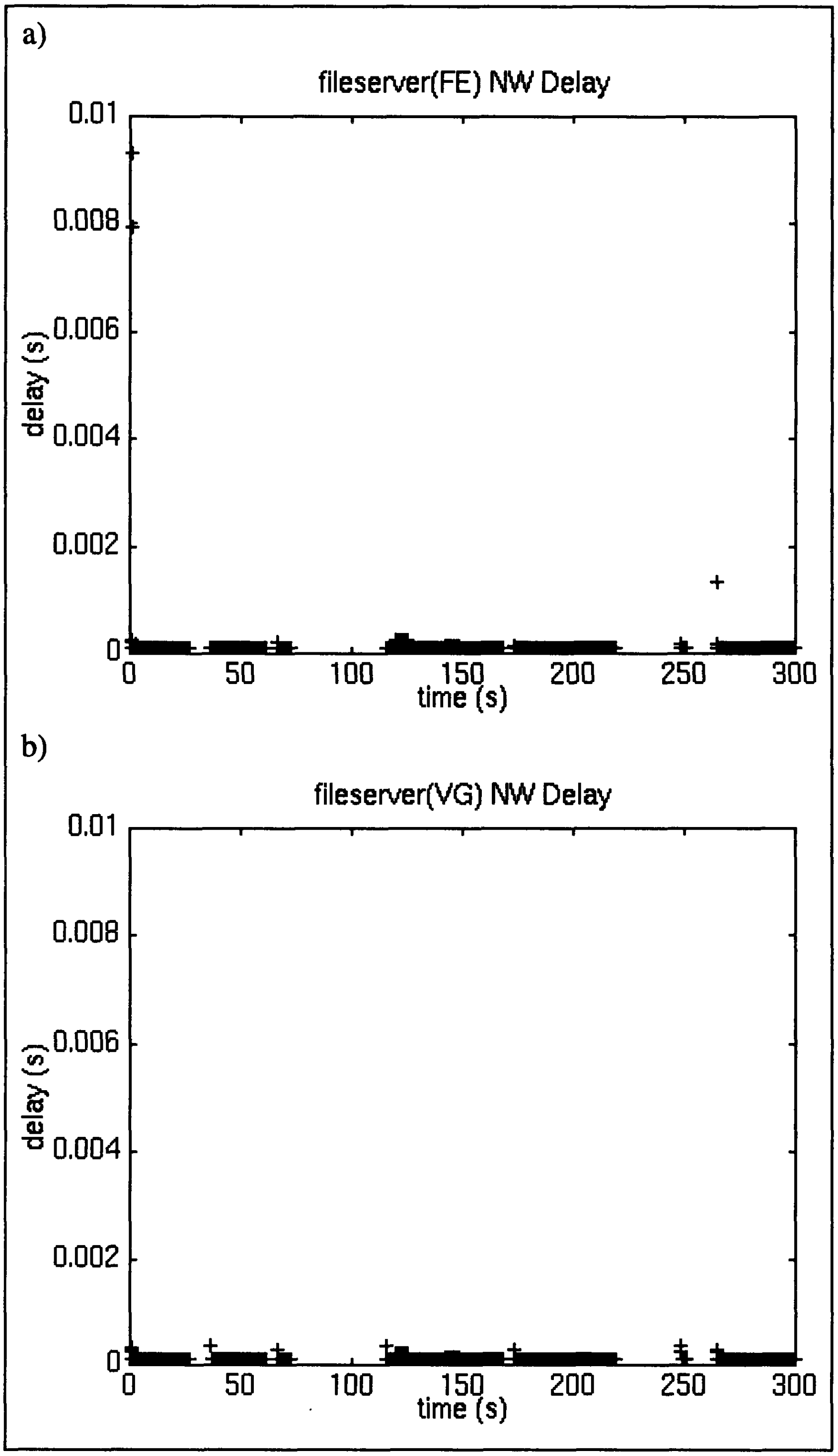

Figure 5.9: The network delay experienced during the simulation of the configuration with 2 US machines, 80 Mbps SCSI-2, and a 512 KB block.

shown on the same scale in order to compare them, although a lot of detail of the network delay on the 100VG network is lost. (The network delay on the 100VG in Figure 5.9(b) is similar to that in Figure 5.4(b) from the previous run.) The buffer on the 100VG ultrasound machine either sees one or zero packets, but never experiences a buildup like the buffers of the ultrasound machines on Fast Ethernet, therefore the network delay times stay smaller on 100VG.

The graphs of the growth of the fileserver buffer for this simulation, Figure 5.10, also show the effect of increasing the block size. Because of the large block size, the fileserver is able to get more data to the disk, and is actually able to clear its buffer at times. In comparison to the buffer sizes from the previous simulation (Figure 5.5)[1], the max buffer size

1. Please note the difference in scale between the two graphs of the fileserver buffer size. When using the small 4KB block, the buffer size in Figure 5.5 is graphed on a scale from 0 to 3×10^8 bytes, whereas the Figure 5.10 corresponding to the use of the 512KB block is graphed on a scale from 0 to 6×10^5 bytes.

in this case is about 500 times smaller. Obviously, increasing the block size has a tremendous effect on the fileserver.

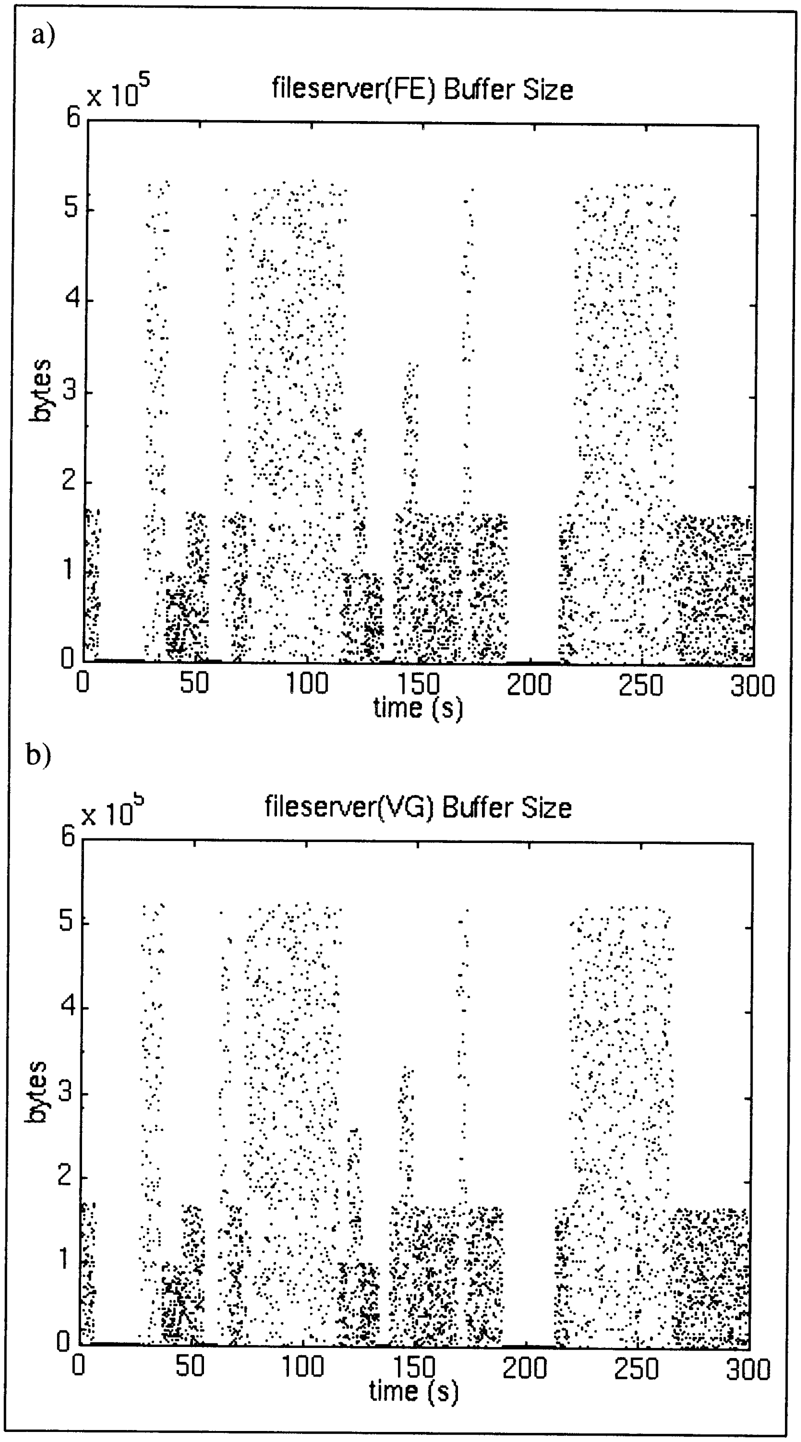

Figure 5.10: The growth of the fileserver buffer from the simulation of the configuration involving 2 US machines, 80 Mbps SCSI-2, and a block size of 512KB.

The use of the larger block size also greatly reduces the delay time that the information experiences while waiting at the fileserver to be written to the disk, as shown in Figure 5.11.

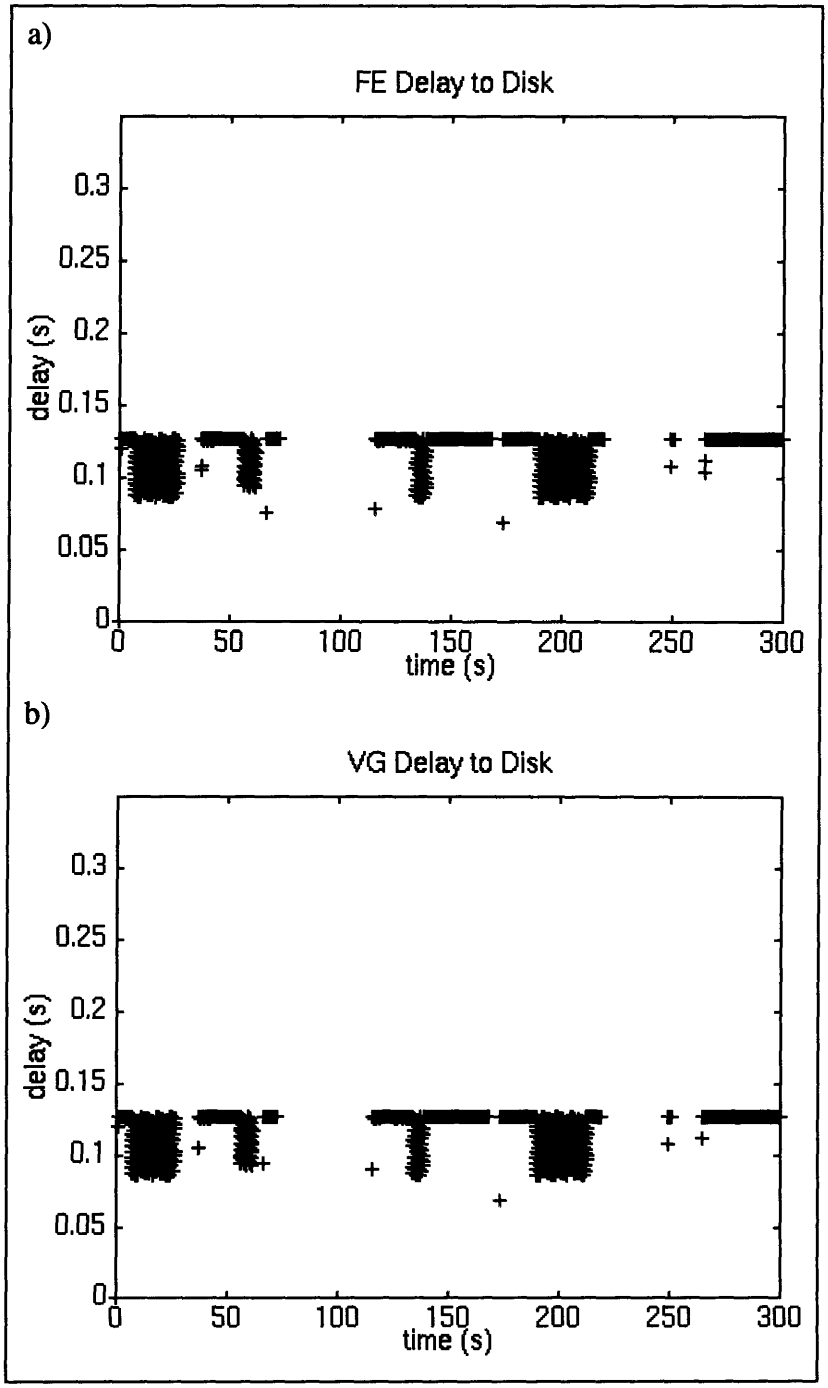

Figure 5.11: The delay to disk for the simulation with 2 US machines, 80 Mbps SCSI-2, and a 512KB block.

A larger block size is also a boon to the reviewstation, evident from Figure 5.12, which shows how the reviewstation receives data over the course of the simulation. In this run

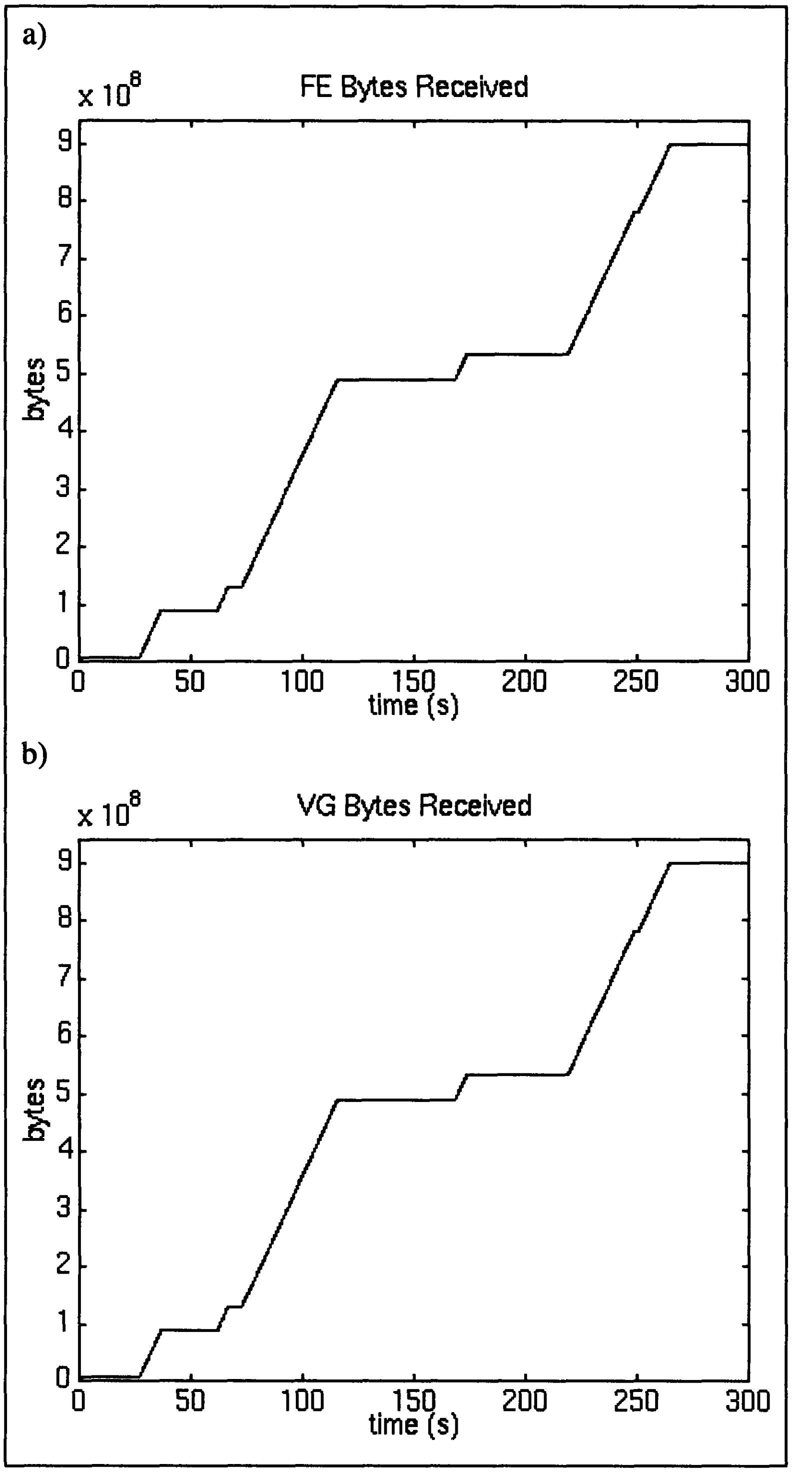

Figure 5.12: The bytes received by the reviewstation for the simulation involving 2 US machines, 80 Mbps SCSI-2, and a block size of 512KB.

using the larger 512KB block, the reviewstation was able to receive about 4000 times the amount of data that the reviewstation received in the previous simulation using a small 4KB block for the transfer to the disk.

5.2.3 IMACS configuration with 5 US machines, 80 Mbps SCSI, 4 KB block size

Figure 5.13 shows the offered load from the first ultrasound machine on the network. This machine's offered load over the course of the simulation consists of a few small intervals of 2D interspersed with small gaps, and then later, some very long gaps.

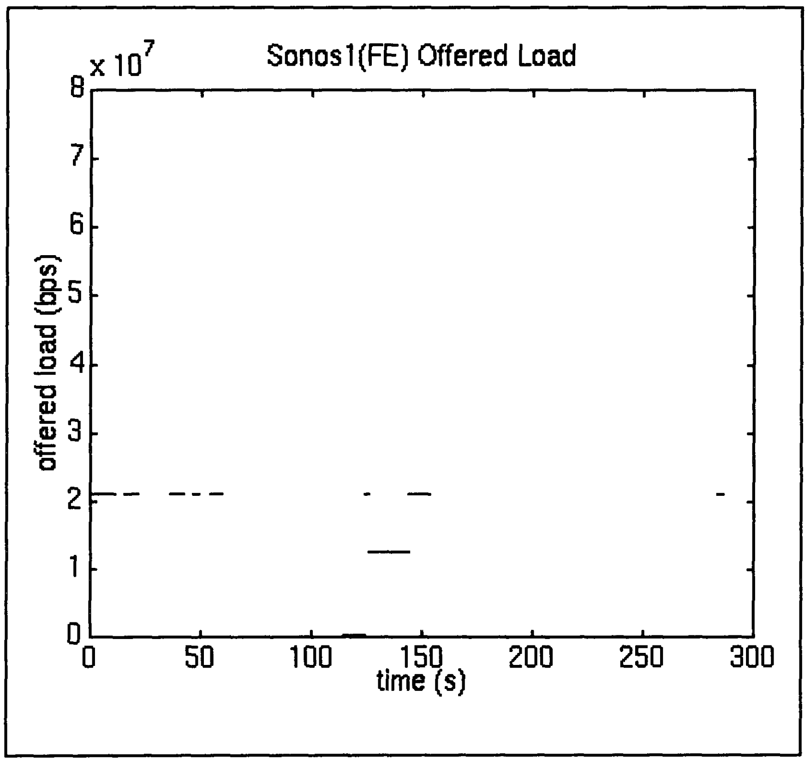

Figure 5.13: The offered load from one of the ultrasound machines in the simulations with 5 ultrasound machines.

In contrast, Figure 5.14 below shows the total offered load placed on the networks (both the 100VG and the Fast Ethernet) by the five ultrasound machines. As the graph

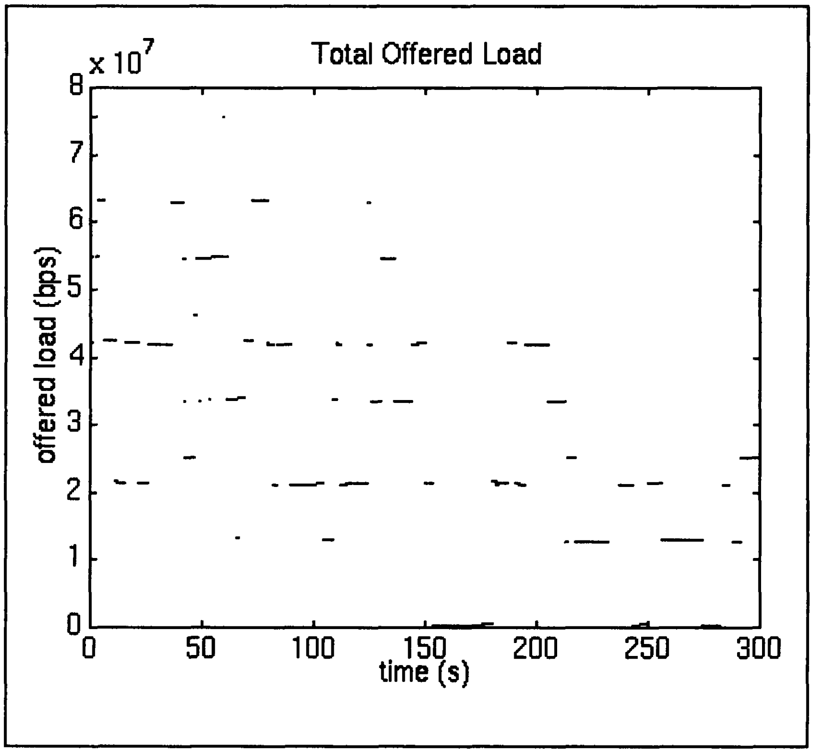

Figure 5.14: The total offered load from the five ultrasound machines.

shows, for this run, there was almost no time when all the machines were generating 2D; and similarly there was almost no time when all the machines were in a gap.

The buffer sizes from one ultrasound machine on both Fast Ethernet and 100VG are shown in Figure 5.15. The graph of the buffer on the Fast Ethernet ultrasound machine shows that a large number of packets can be stored in the machine's buffer due to backoff.

The buffer size graph from the ultrasound machine on the 100VG medium, however, is similar to that from the previous section, never holding more than one packet.

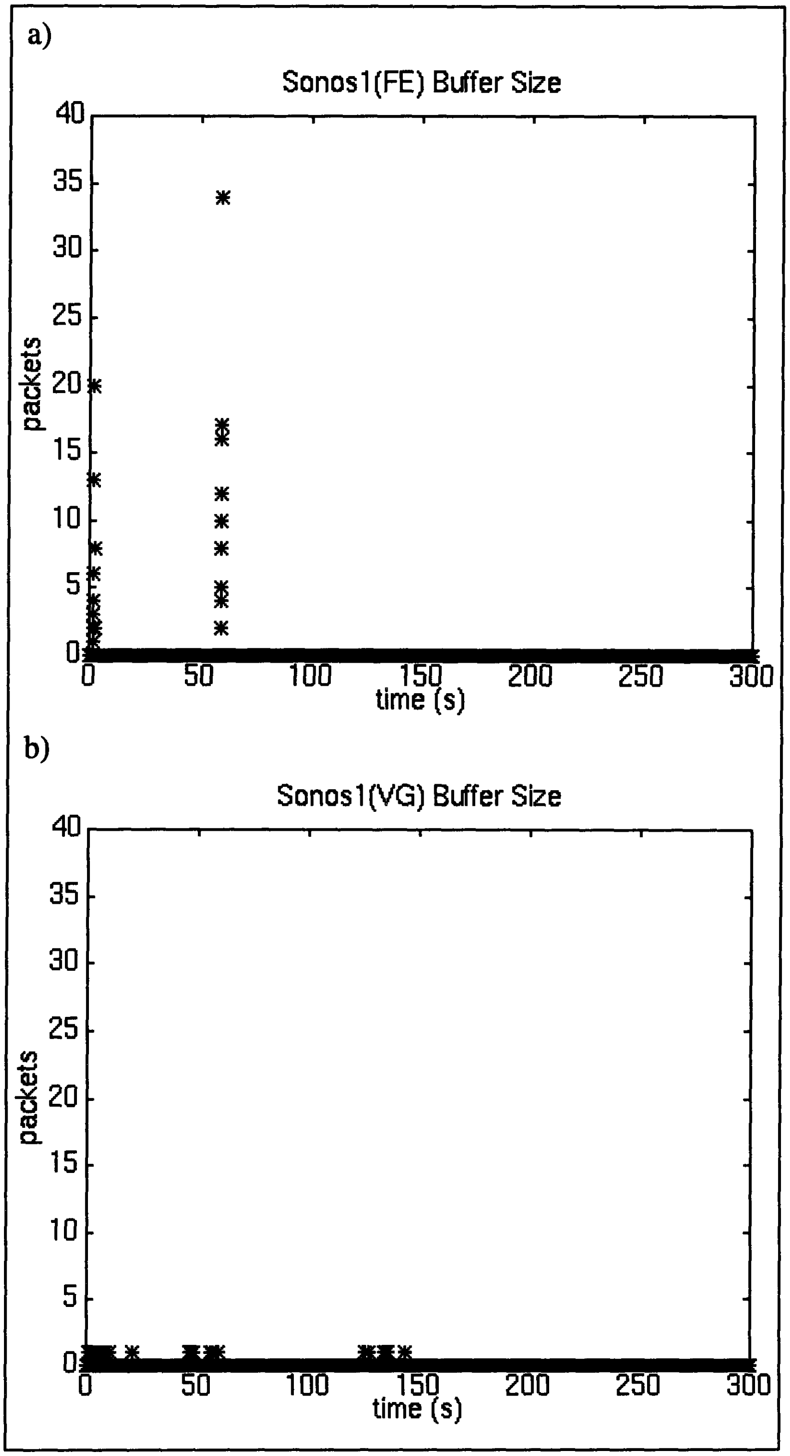

Figure 5.15: The growth of the buffer on one ultrasound machine from the configuration with 5 US machines, normal SCSI-2, and a 4KB block.

The network delay experienced by the packets is shown below in Figure 5.16. Because there are now five machines generating data, the network delay on the Fast Ethernet is much higher than the delay on the 100VG due to backoff.[1].

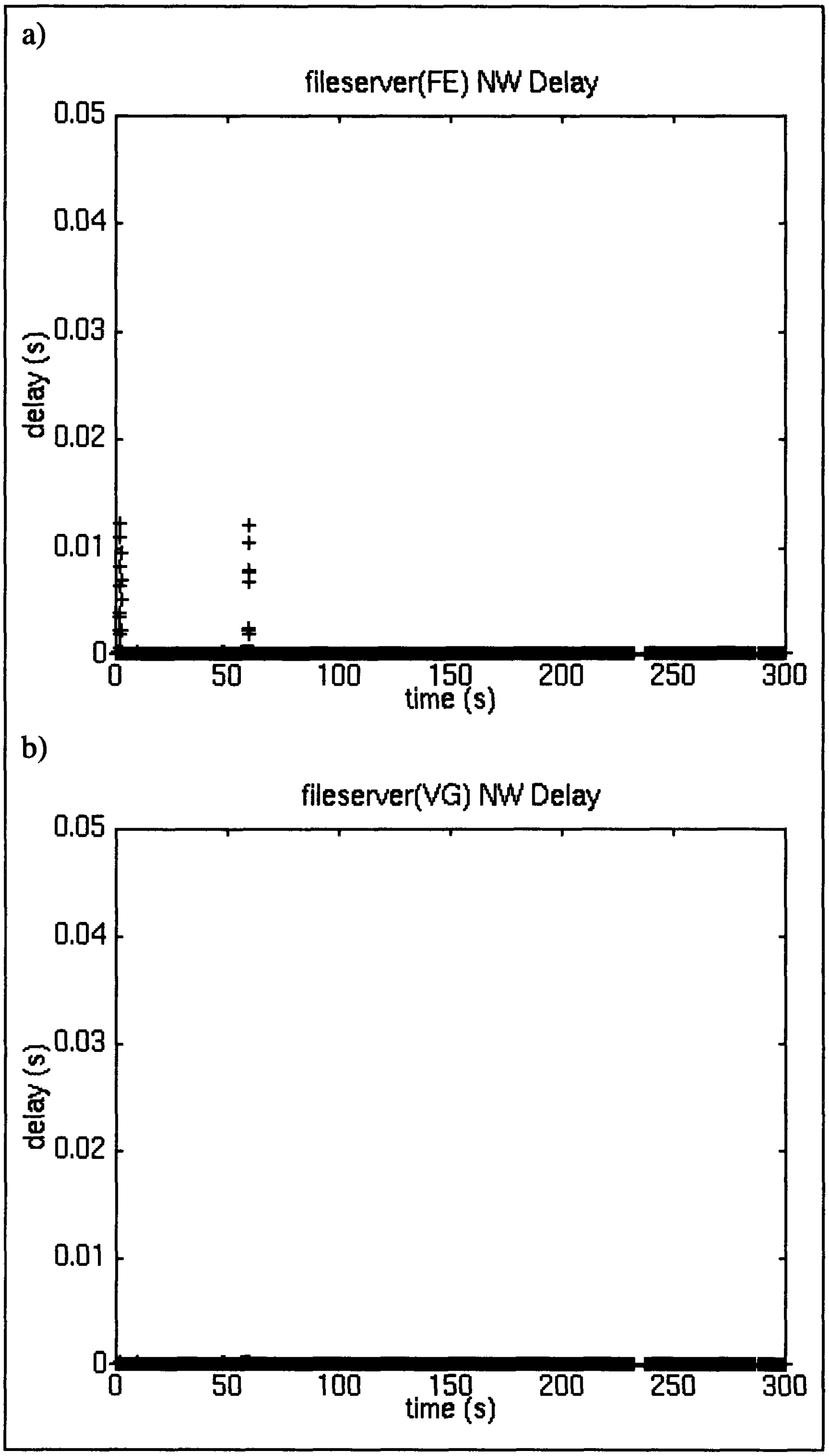

Figure 5.16: The network delay from the simulation involving 5 US machines, normal SCSI-2, and a 4KB block.

Using the small block size of 4KB for the transfer from the fileserver to the disk causes the fileserver buffer to continue to build up for the length of the simulation, and never allows to empty. Even during the two intervals when all the machines are in a gap (Figure 5.14) the fileserver is so overloaded, that its size only slows in growth but never reverses.

1. In order to compare the network delays from all the runs involving 5 ultrasound machines, all the graphs were drawn using a vertical scale from 0 to 0.05 seconds. Unfortunately a good amount of detail is lost, especially for the graph of the delay on the 100VG.

Clearly, again, a large block size is quite necessary in order to keep the memory requirements on the fileserver reasonable.

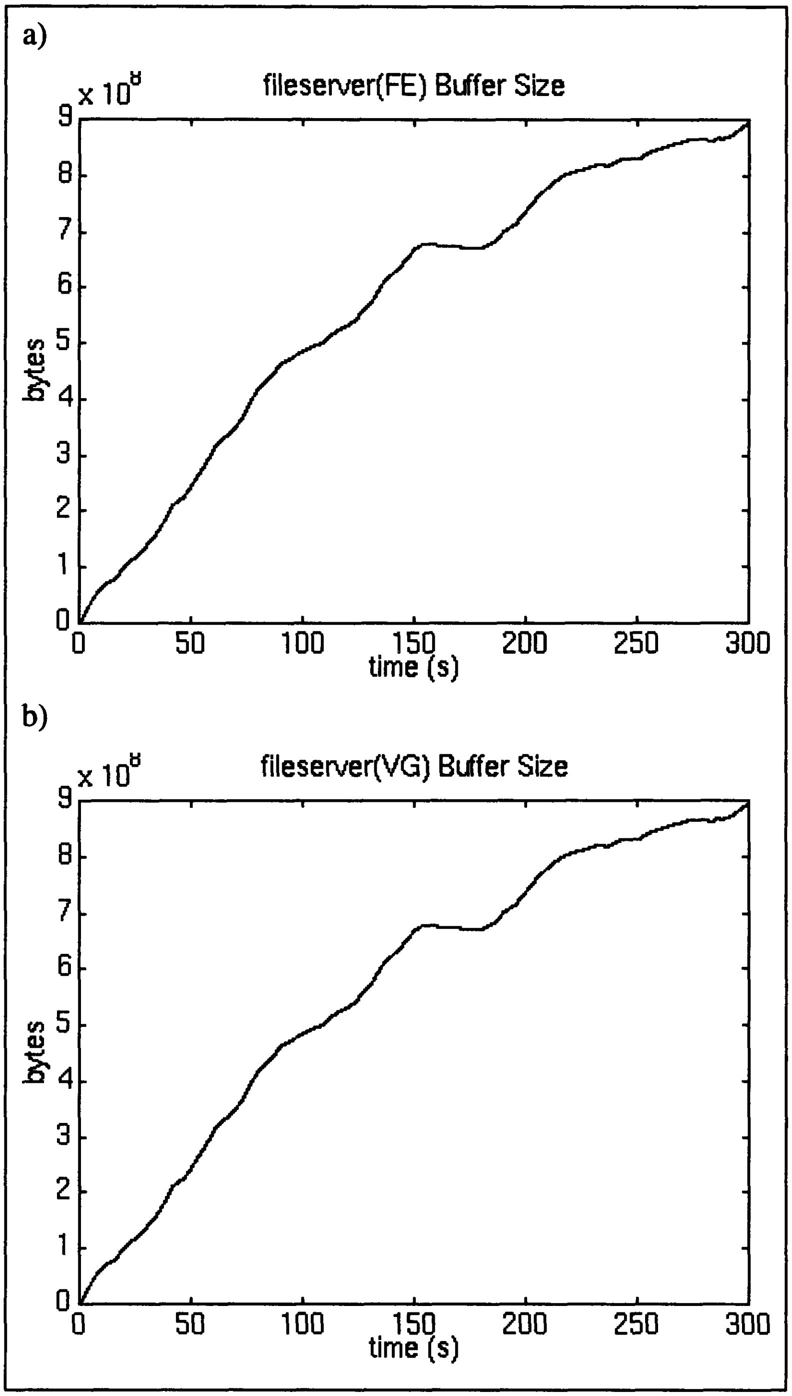

Figure 5.17: The growth of the fileserver buffer from the configuration with 5 US, normal SCSI-2, and a 4KB block.

The reason for the buildup can be seen from the delay that the information experiences while waiting to be written to the disk. Data comes in to the fileserver at a much faster rate than it can be written out since each write to the disk incurs the hit of the disk seek time, and data is being written in such small blocks. Thus the time between when a packet arrives at the fileserver, and when it can be written to the disk grows, since each packet must wait before all of the ones that have arrived before it get written to the disk. Obviously, in order to get the data written to the disk in a more timely manner, a much larger block size must be used. Using a larger block size such as the 512KB block will also greatly alleviate the burden on the fileserver buffer.

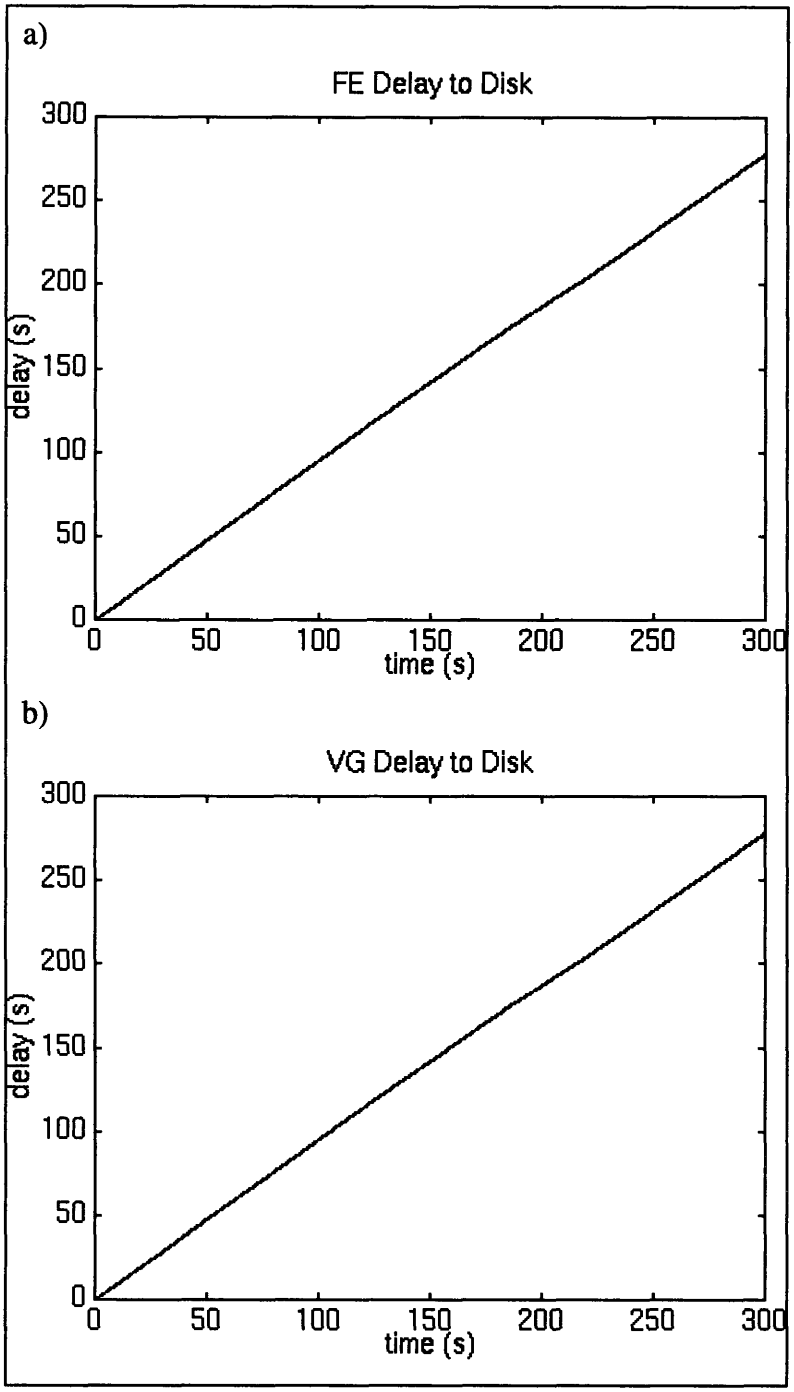

Figure 5.18: The delay to disk from the run of the configuration with 5 US, normal SCSI-2, and a 4KB block.

In addition to causing the fileserver buffer to grow tremendously, the use of the small block size for the transfer to the disk also prevents the fileserver from servicing the

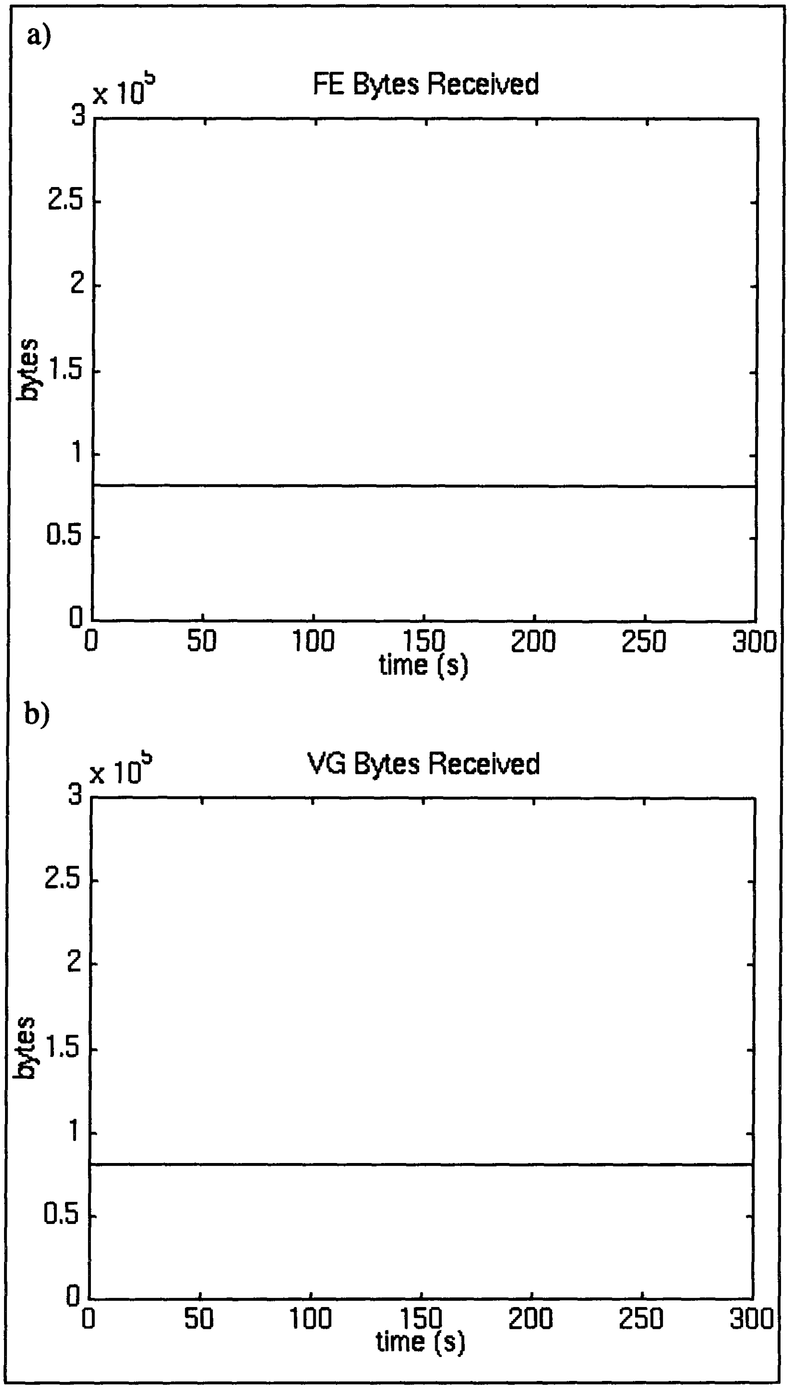

Figure 5.19: The bytes received by the reviewstation from the run of the configuration with 5 US, normal SCSI-2, and a block size of 4 KB.

reviewstation's request for exam data, even during gaps, since the policy it follows is to give preference to writes to the disk, and it always has data to write to the disk. The data received by the reviewstation over time is shown in Figure 5.19.

5.2.4 IMACS configuration with 5 US machines, 80 Mbps SCSI, 512 KB block size

The simulation run of the configuration with 5 ultrasound machines, an 80 Mbps SCSI-2 bus, and a block size of 512KB experienced the exact same offered load from the ultrasound machines as the run discussed in the previous subsection. Therefore, those graphs will not be reshown here.

The buffer sizes of the ultrasound machines are shown in Figure 5.20. Overall, the graphs are similar to those from the previous section (Figure 5.15), but the use of the larger block size does have an effect on the ultrasound machine on Fast Ethernet since the fileserver is able to add network load by sending data to the reviewstation. The added load causes the machine to buffer more packets than it would if the fileserver were not able to send on the network. The 100VG buffer, however, still does not experience a buildup of packets.

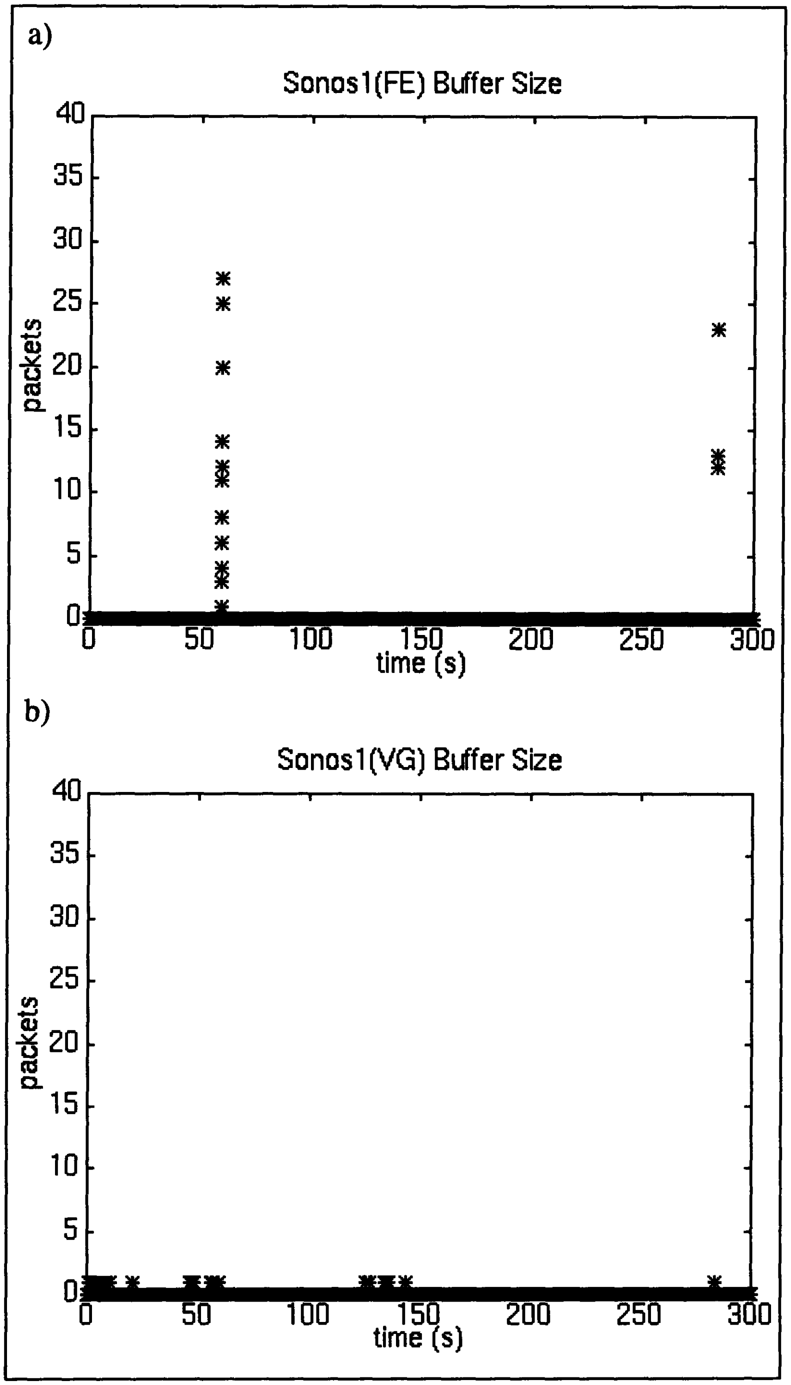

Figure 5.20: The growth of the buffer on one ultrasound machine from the configuration with 5 US machines, normal SCSI-2, and a 512KB block.

The network delay experienced by the packets for this simulation run is shown below in Figure 5.21. The network delay is slightly different from the delay from the previous run (Figure 5.16), because there is slightly more load on the network since the fileserver is able to send data to the reviewstation.

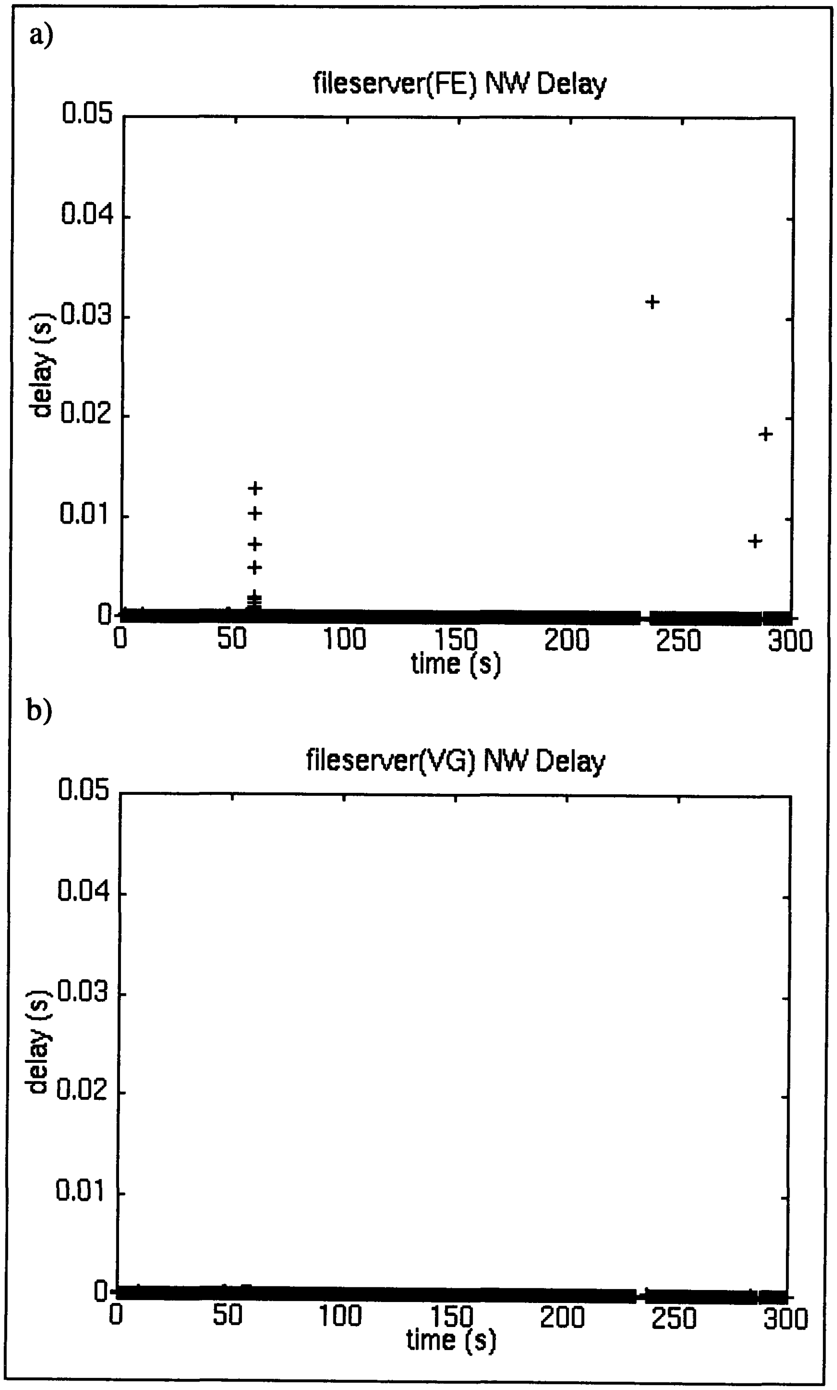

Figure 5.21: The network delay from the simulation involving 5 US machines, normal SCSI-2, and a 512KB block.

Just as in the case with two ultrasound machines and a large block, the fileserver buffer is greatly affected by the change to the 512KB block size for the transfer to the disk, as it should be.

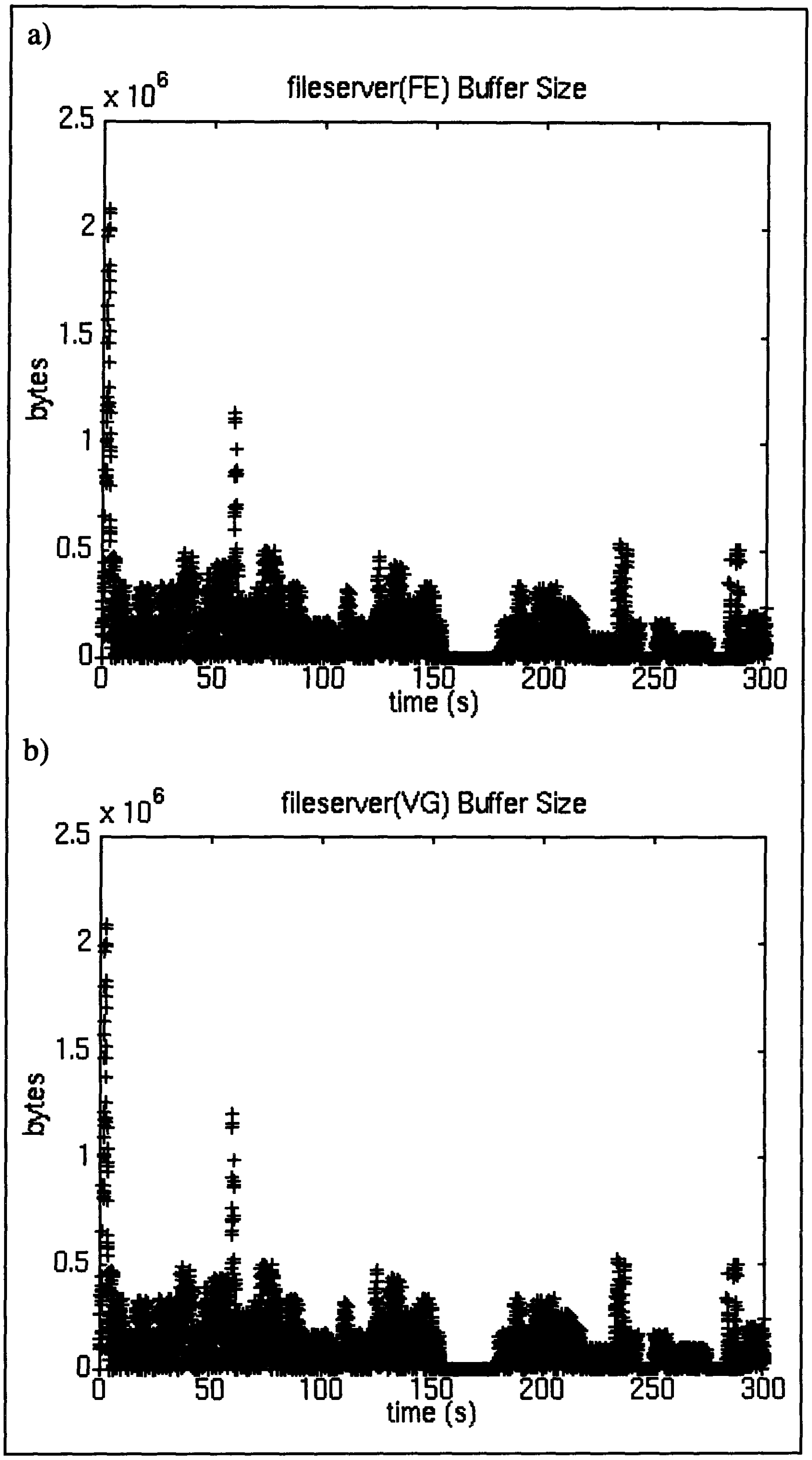

Figure 5.22: The growth of the fileserver buffer from the configuration with 5 US, normal SCSI-2, and a block size of 512KB.

The delay to disk, in Figure 5.23, is also significantly affected by the change to the 512KB block.

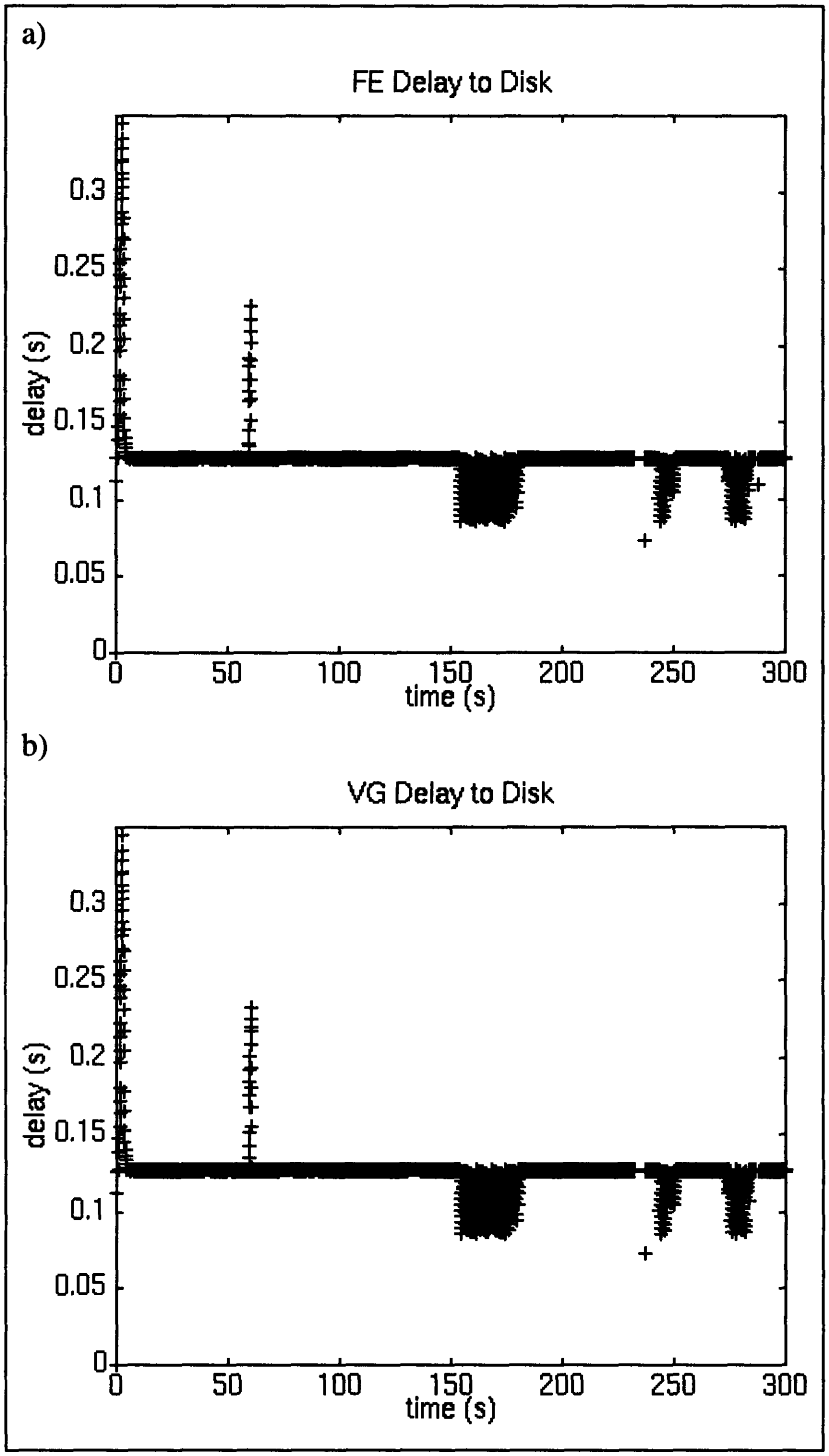

Figure 5.23: The delay to disk from the run of the configuration with 5 US, normal SCSI-2, and a block size of 512KB.

The exam data received by the reviewstation over the course of the simulation is shown in Figure 5.24. Just as in the case with two ultrasound machines discussed earlier,

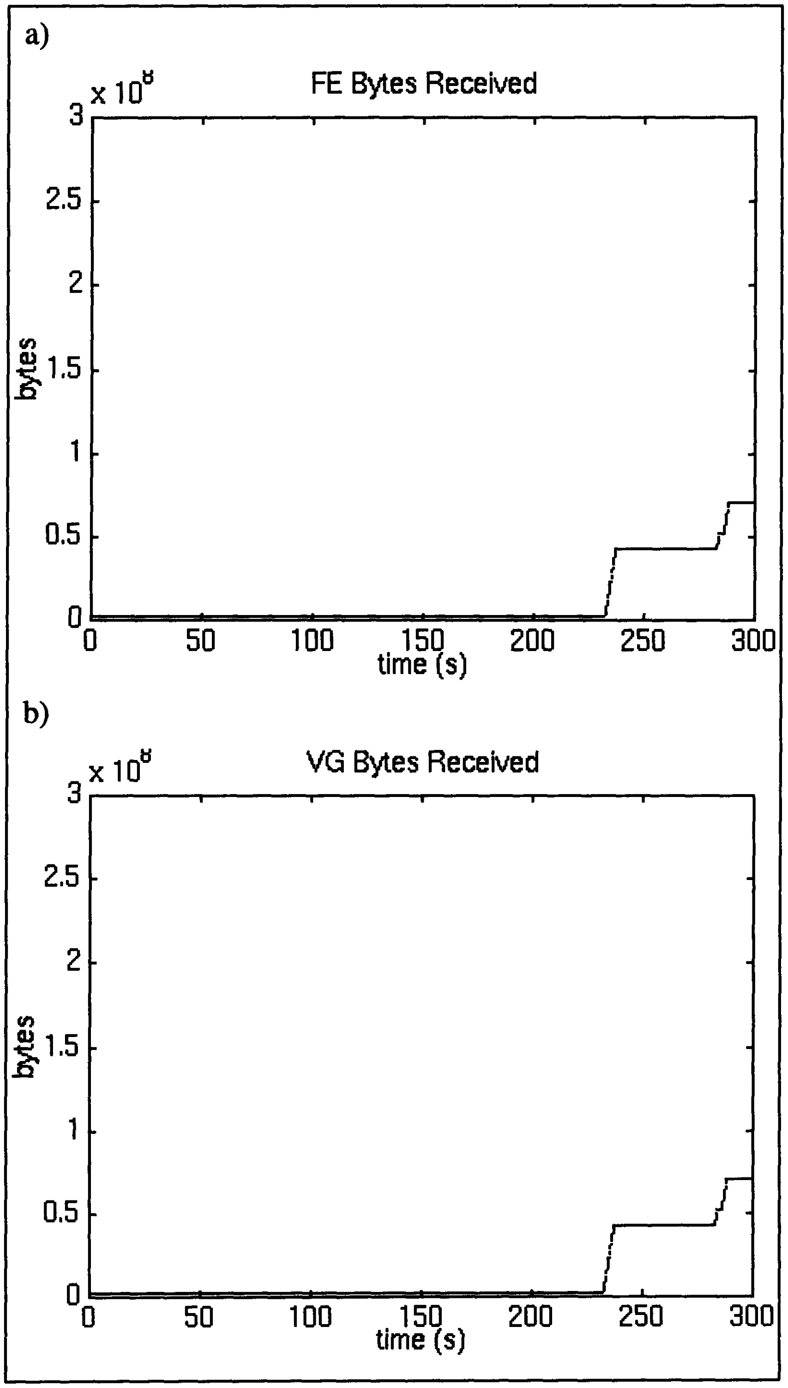

Figure 5.24: The bytes received by the reviewstation from the run of the configuration with 5 US, normal SCSI-2, and a block size of 512KB.

the reviewstation is actually able to get data back when using a large block, in contrast to what it receives when using a small block (Figure 5.19).

5.2.5 IMACS configuration with 5 US machines, 160 Mbps SCSI-2, 512 KB block

The simulation run of the configuration with 5 ultrasound machines, a 160 Mbps SCSI-2 bus, and a block size of 512KB experienced the exact same offered load from the ultrasound machines as the previous two configurations simulated, so those graphs will also not be reshown here.

The graph in Figure 5.25 shows the growth of the buffer on the first ultrasound machine on both 100VG and Fast Ethernet. Both graphs are scaled to accommodate the maximum value found when sampling the ultrasound machine buffer in the simulation using the Fast Ethernet model. The buffer on the machine in the simulation using the 100VG network model appears to either hold one packet or be empty, whereas the buffer on the Fast Ethernet ultrasound machine really builds up at times. The first two spikes in the growth of the Sonos1(FE) buffer correspond to times when that machine was generating 2D data, and the other machines were also placing a high load on the network. The third spike corresponds to a time when the fileserver was attempting to send a lot of data to the reviewstation over the network.

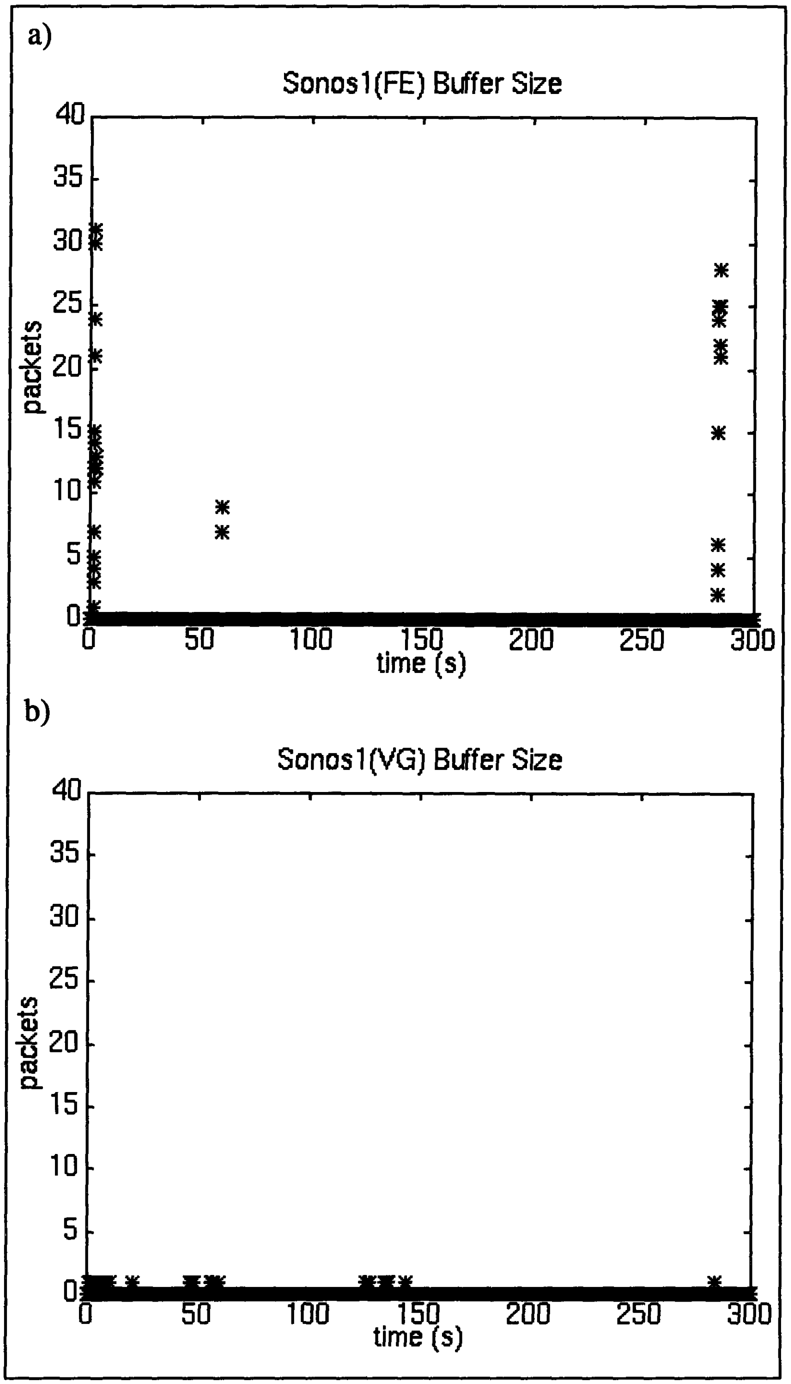

Figure 5.25: The growth of the buffer on one of the five ultrasound machines on a)Fast Ethernet and b)100VG from the simulation involving fast-wide SCSI-2 and a block size of 512KB.

The next pair of graphs shows the network delay that the ethernet packets experienced. The network delay is the difference between the creation time at the ultrasound machine, and the time the packet is received by the fileserver at the other end of the network. Here,

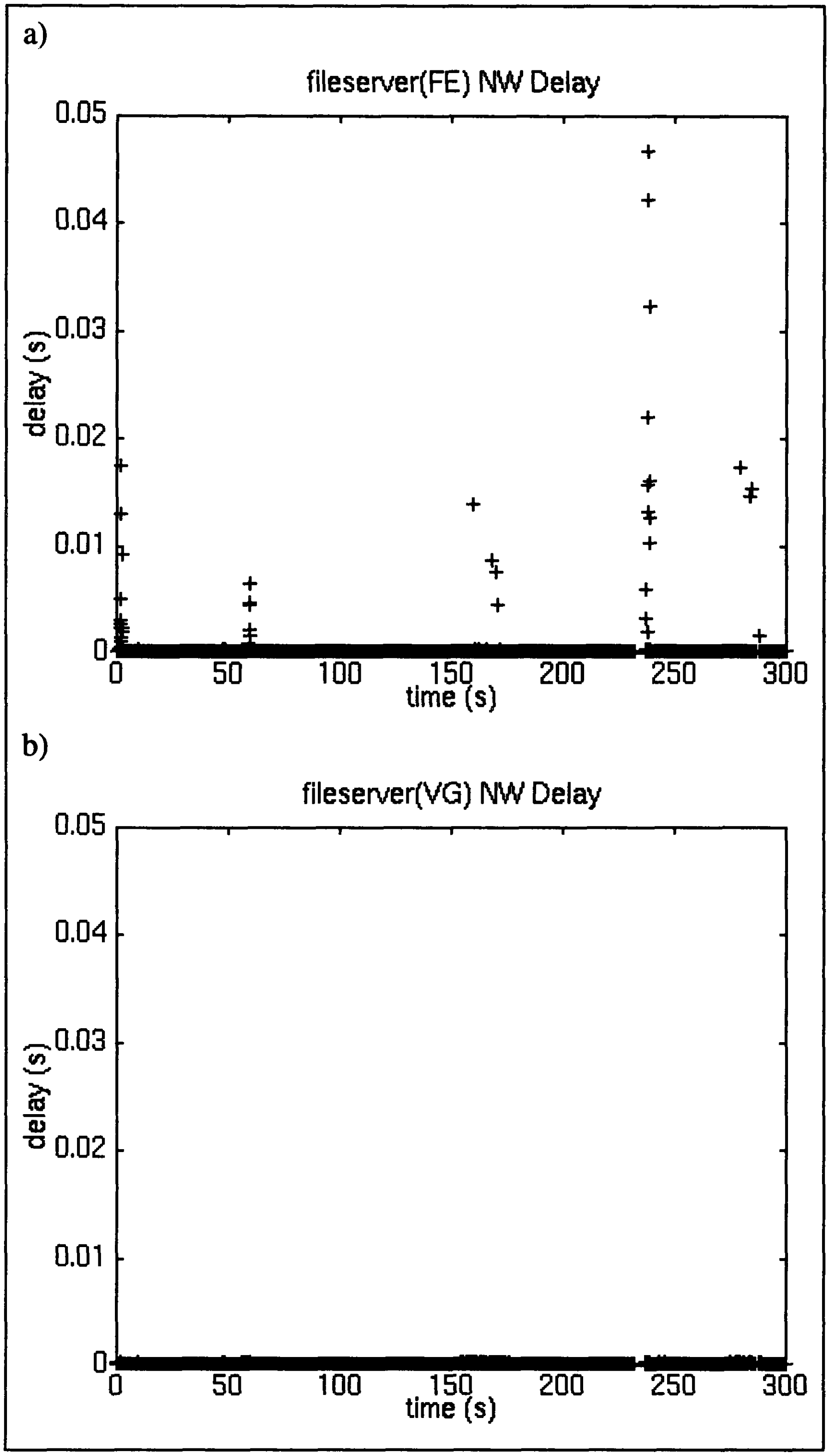

Figure 5.26: The network delay on a)Fast Ethernet and b)100VG from the simulation involving fast-wide SCSI-2 and a block size of 512KB.

again, the network delay on the Fast Ethernet is much higher than on the 100VG. The highest spike in the graph of the delay times on the Fast Ethernet around 240 seconds, occurs at the same time that the reviewstation is receiving data from the fileserver.

The growth of the fileserver buffer for the runs for this configuration of five ultrasound machines, using fast-wide SCSI-2 with a large block size can be seen in Figure 5.27. Overall, because of the large block size of 512 KB, and the use of the fast-wide SCSI-2 bus, the fileserver really has no problem getting rid of the data it receives from the ultrasound machines to the disk. The problem comes, however, when it receives back a lot of data from the disk, but is unable to send it on the network with the same speed at which it receives it. The two spikes in the following graph correspond to times when the fileserver is servicing the request for exam data from the reviewstation. The fileserver buffer on the Fast Ethernet system was bigger than the fileserver buffer on the 100VG system when the fileserver was trying to send data to the reviewstation, probably because it had to occasionally backoff in vying for the network with the ultrasound machines that were also trying to send data to the fileserver. In comparison to the growth of the fileserver buffer from

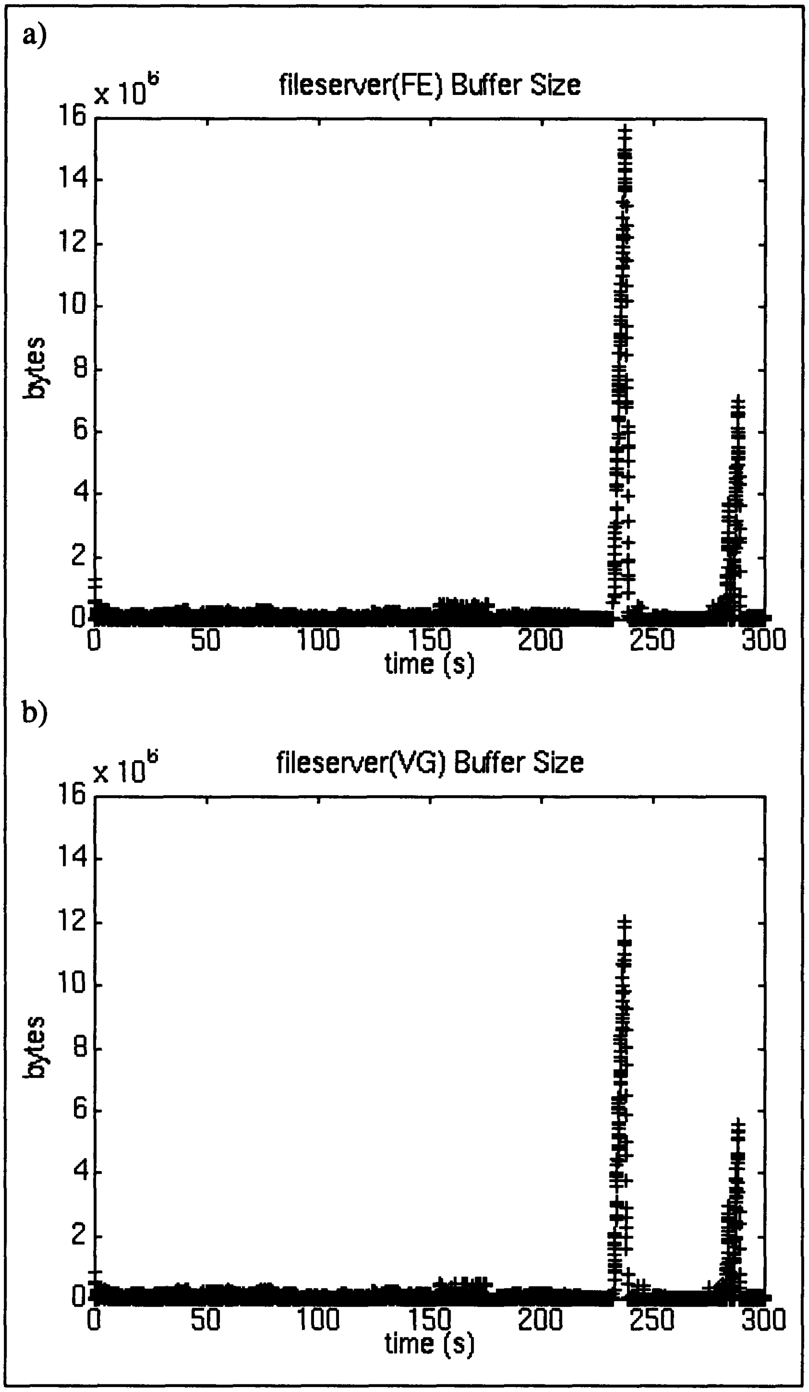

Figure 5.27: The growth of the fileserver buffer for the configuration involving 5 ultrasound machines, fast-wide SCSI-2, and a large block size.

the configuration using the slower 80 Mbps SCSI-2, Figure 5.22, the fileserver buffer size in Figure 5.27 appears to be bigger most of the time. Since the speed of the bus in this run is double that of the bus in the previous run, the fileserver is able to get much more data to the disk over time, therefore the buffer is usually smaller.

The next pair of graphs show the delay that the information from the ultrasound machines experiences between arriving at the fileserver and being written to the disk. Most of the time, there is a relatively high delay because the packets are coming in to the fileserver faster than they are being written to the disk. During the interval approximately between 150 and 200 seconds, the delay times really drop. This happens because 4 of the ultrasound machines are not generating any data, and one is generating Doppler data, so the fileserver is getting the data to write very slowly, and as a result is writing short blocks to the disk.

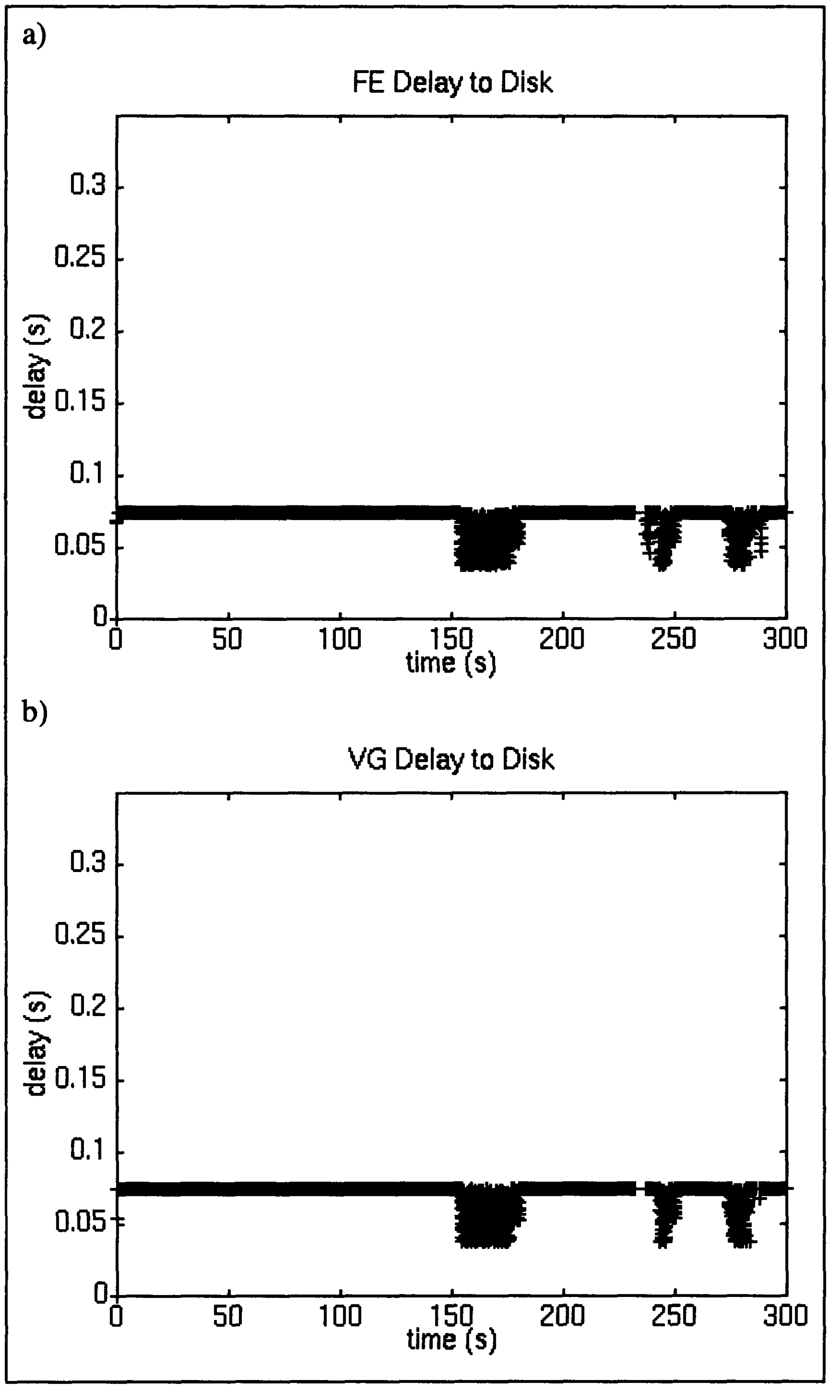

Figure 5.28: The delay to disk from the simulation involving 5 ultrasound machines, fast-wide SCSI-2, and a block size of 512 KB.

Finally, it is important to see how the reviewstation receives data back from the fileserver over time. Because the fileserver gives preference to writing to the disk over reading from it, it is only able to service the request for data when it has nothing to write to the disk, which happens either when none of the machines are generating data, or when the machines are generating data very slowly. As Figure 5.29 shows, the reviewstation is only able to receive data back at three times. The first time when the reviewstation receives data corresponds to a time (from Figure 5.14) when 4 of the ultrasound machines were not generating any data, and one was generating Doppler data. The other two times correspond to gaps in the total offered load on the system from the ultrasound machines.

Figure 5.29 also shows an interesting difference between the use of the switched 100VG and the shared Fast Ethernet. The reviewstation on the Fast Ethernet received more data from the fileserver by the end of the 5 minutes of simulation than did the reviewstation on the 100VG. Aside from the interval shortly after 150 seconds, both machines receive data back at the same rate. During that interval, however, the reviewstation on the Fast Ethernet manages to receive more than its 100VG counterpart. The reason this happens is that during that time, four of the ultrasound machines were generating nothing, but one was generating Doppler. The 100VG by virtue of its Demand Priority transmission control policy, serviced both the fileserver sending data to the reviewstation, and the ultrasound machine sending data to the fileserver equally. On the Fast Ethernet, however, the machine generating Doppler would see the network as busy because the fileserver was sending data to the reviewstation. Since the Doppler data generation rate is so slow, the fileserver buffer would clear and allow the servicing of the reviewstation's request for exam data. During this interval, the fileserver effectively gets the Fast Ethernet all to itself, whereas the fileserver on the 100VG network must share the hub evenly with the only machine generating.

It may appear from the graphs that the use of the faster bus does appear to have a significant effect, however, since this run using the faster SCSI bus saw the reviewstation

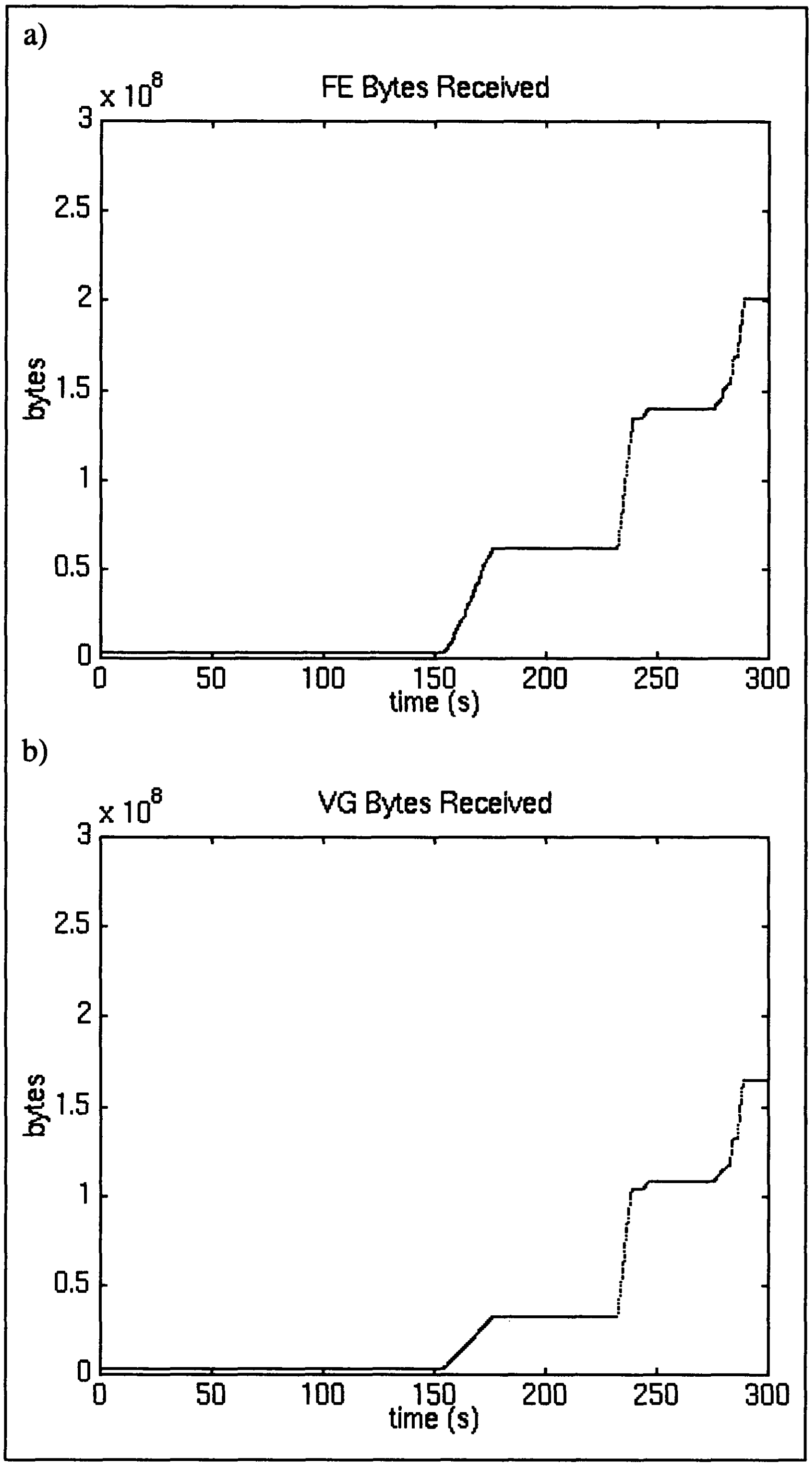

Figure 5.29: The bytes received by the reviewstation over the course of the simulation of the configuration with 5 US, 160 Mbps SCSI-2, and a block size of 512KB.

receive much more data than with the slower bus (Figure 5.24), primarily because it receives data back at three different times instead of two as in these graphs. Since the fileserver model does follow a policy of servicing the reviewstation's request for exam data only when its buffer has been cleared, having a shorter time to wait between disk accesses gives the fileserver a better chance of writing its buffer to the disk, waiting for the disk to finish, and then finding that it has nothing more to write, so it can service the reviewstation's request. But, in this case, only one machine (Figure 5.14) was generating during the interval shortly after 150 seconds when the reviewstation on the configuration using the faster SCSI-2 bus was receiving information (Figure 5.29). The fileserver was writing to the disk fast enough to be able to service the reviewstation only during this time, but not any other, showing that the total load during all the other times (save the times when all machines were in a gap) was too much for the fileserver to be able to service the reviewstation more often. Therefore, simply doubling the SCSI bus speed is not enough to allow the fileserver to service the reviewstation's request for exam data. Changing how the fileserver works is more important.

In order to have the reviewstation receive exams quicker, it is necessary that the fileserver not follow a policy of giving absolute priority to writes to the disk. It would be better to follow a policy of mixing writes and reads to allow both to be serviced better, though not necessarily fairly because preference should still be given to the writes to disk since they cause the fileserver buffer to grow greatly. It would also be better for the fileserver not to send short blocks to the disk, if it does not currently have enough for a block, but instead wait for more data to come in from the ultrasound machines. While the fileserver is waiting to fill these short blocks, it could service the reviewstation's request for exam data.

5.2.6 IMACS configuration with 7 US machines, 160 Mbps SCSI, 512 KB block size

The simulation runs of the configuration with seven ultrasound machines, fast-wide SCSI-2, and a 512KB block used the same seed as the runs having only five ultrasound machines. Because there are more ultrasound machines in this simulation, the total load is different, as is the offered load from each ultrasound machine. The offered load from one ultrasound machine is shown in Figure 5.30.

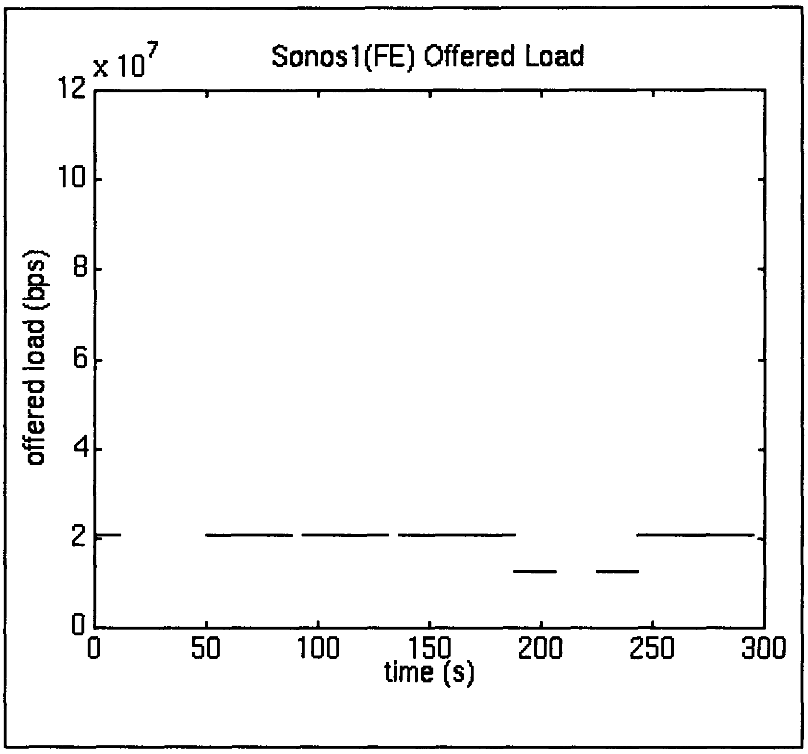

Figure 5.30: The offered load from Sonos #1 in the simulation with 7 ultrasound machines.

The total load placed on the system is shown in Figure 5.31. Although it is possible that all seven of the machines would be generating 2D data at the same time, it never happens over the course of the simulation's five minute duration; although the opposite, when all machines are in a gap, does happen once.

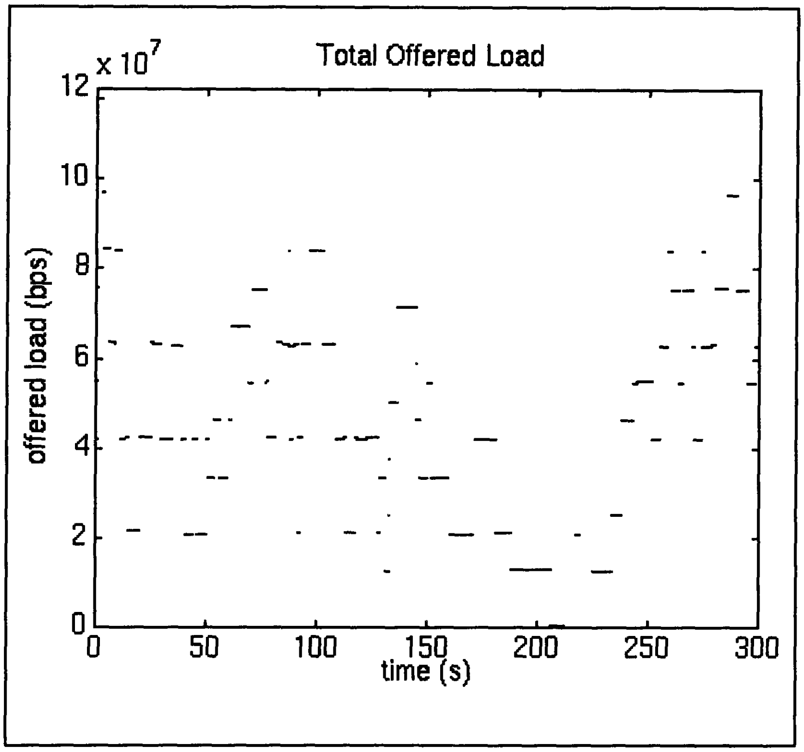

Figure 5.31: The total load placed on the network by the 7 ultrasound machines.

In comparison to the run (in Section) with the same configuration, but five ultrasound machines instead of seven, the buffer on the ultrasound machines do not seem to ever build up on the 100VG, but in this run, the buffer on Sonos1 does build up considerably right at the beginning of the simulation when the machine is generating 2D. The most reasonable explanation for this is that since the machines are generating data periodically, if there is a collision on the first packet, there will be a collision on every other packet during the interval in which the machine is generating. After going through some other generation modes, the machines which were generating packets at the exact same times become out of phase with each other, so no such buildups happen again. The growth of the buffer on Sonos1 for the simulation run with seven ultrasound machines is shown in Figure 5.32.

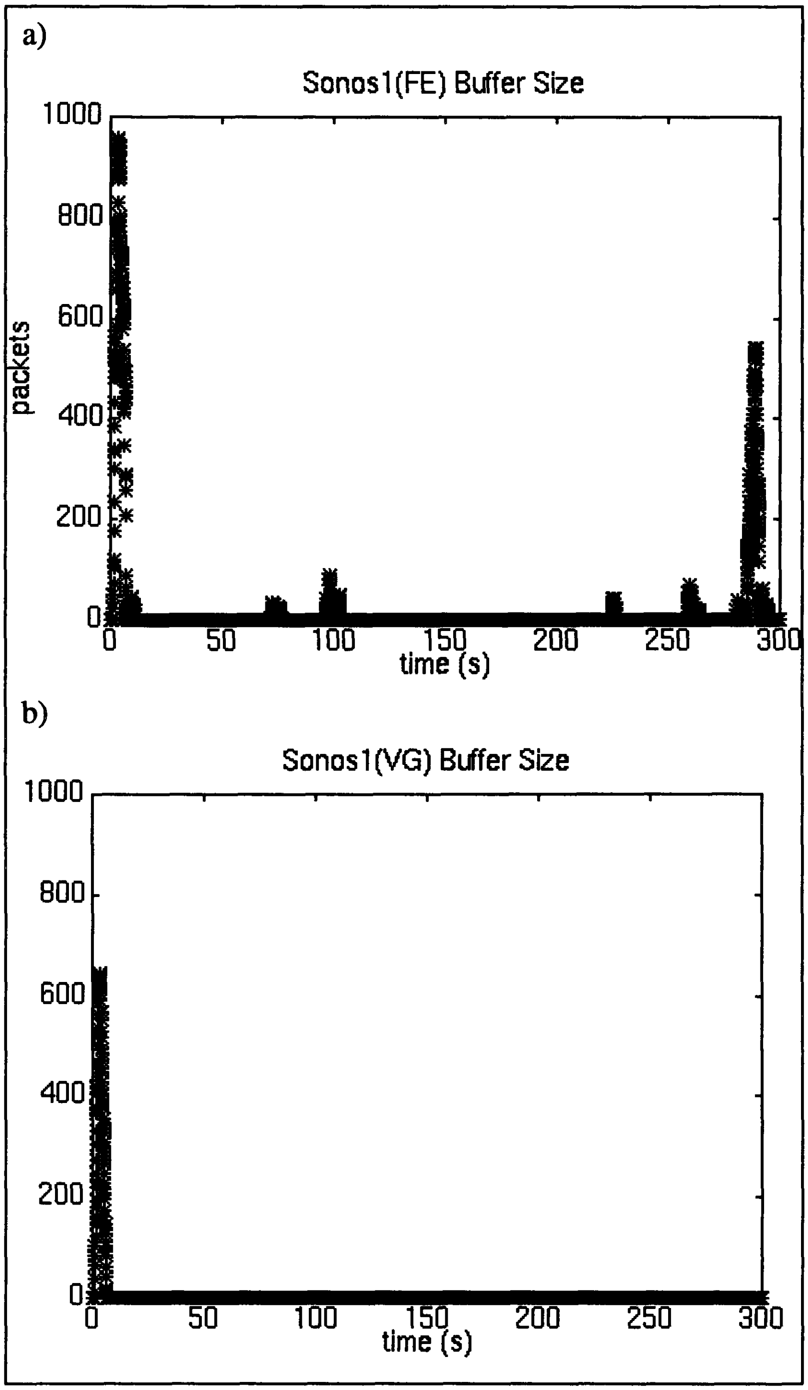

Figure 5.32: The growth of the buffer on one ultrasound machine from the configuration with 7 US machines, fast-wide SCSI-2, and a 512KB block.

The network delay encountered in this simulation run is shown in Figure 5.33. Again, the delay on the Fast Ethernet is greater than the delay on the 100VG, although in this run, the 100VG network delay did spike up right at the beginning of the simulation, corresponding to the buildup of the buffer on Sonos1.

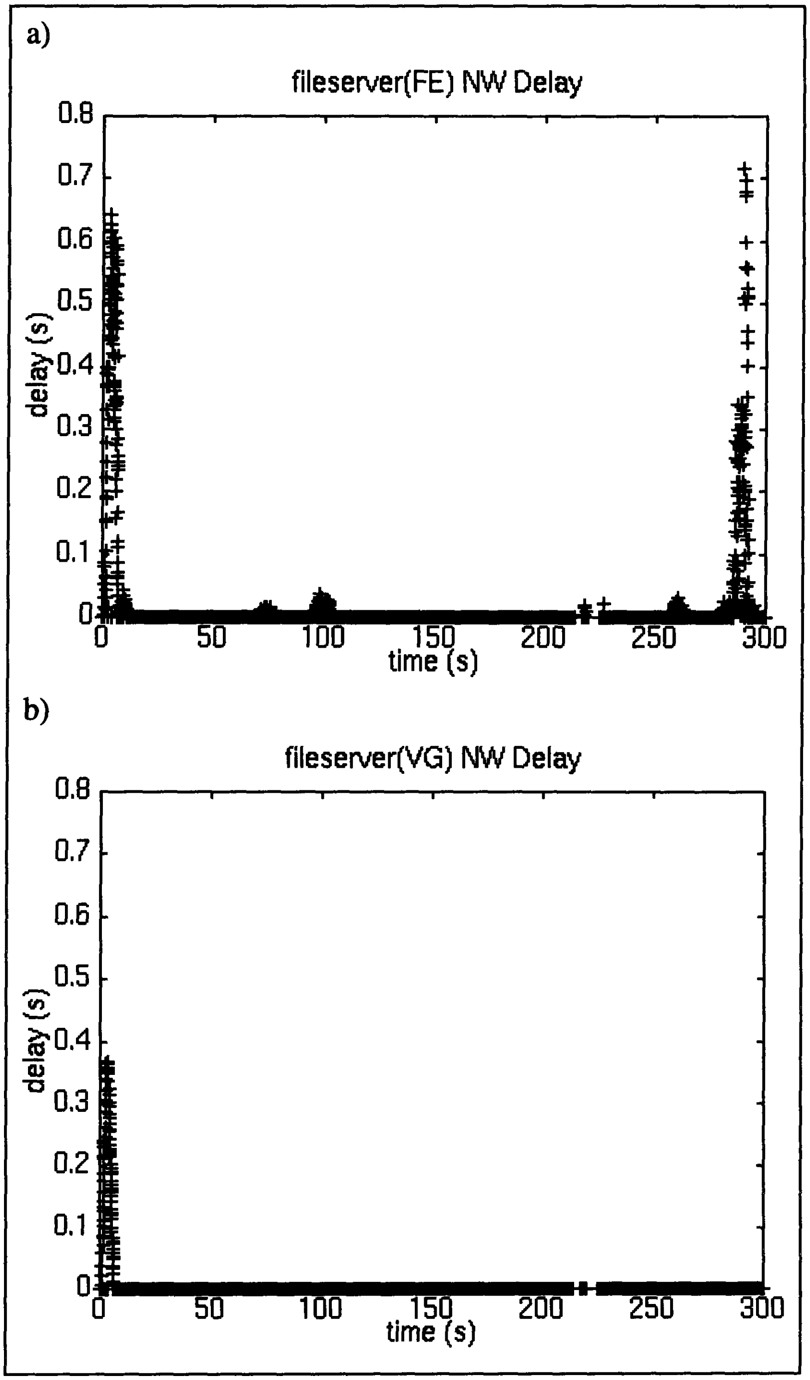

Figure 5.33: The network delay from the run with seven ultrasound machines.

The growth of the fileserver buffer is shown in Figure 5.34. Both graphs have a spike shortly after 200 seconds which corresponds to the time that the fileserver is sending data to the reviewstation, during the only gap which happens during the five minute simulation, when none of the ultrasound machines are generating.

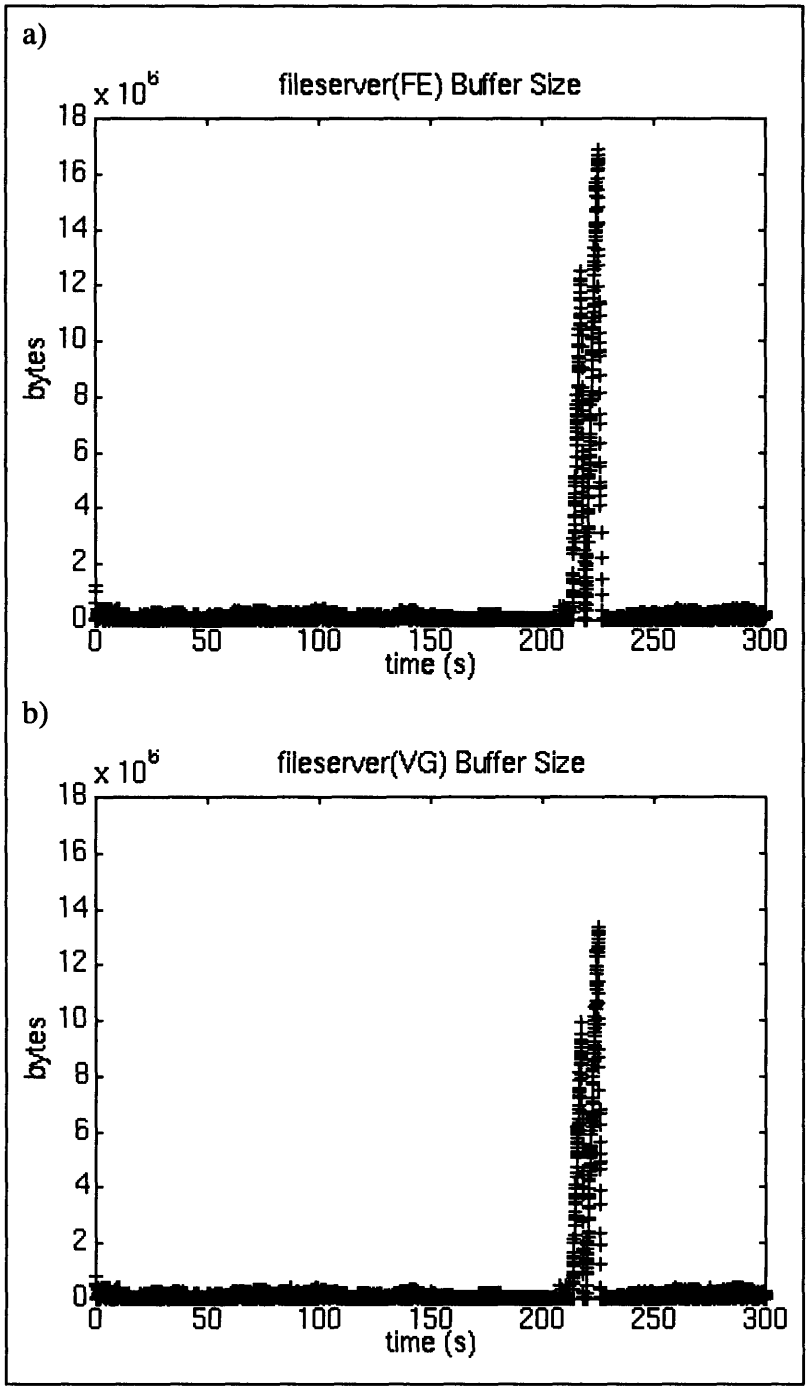

Figure 5.34: The growth of the fileserver buffer from the configuration with 7 US, fast-wide SCSI-2, and a 512KB block.

The delay to disk for this run is shown below in Figure 5.35. The delay is greater than the delay to disk found from the runs with only five ultrasound machines simply because much more data is coming in to the fileserver with seven machines.

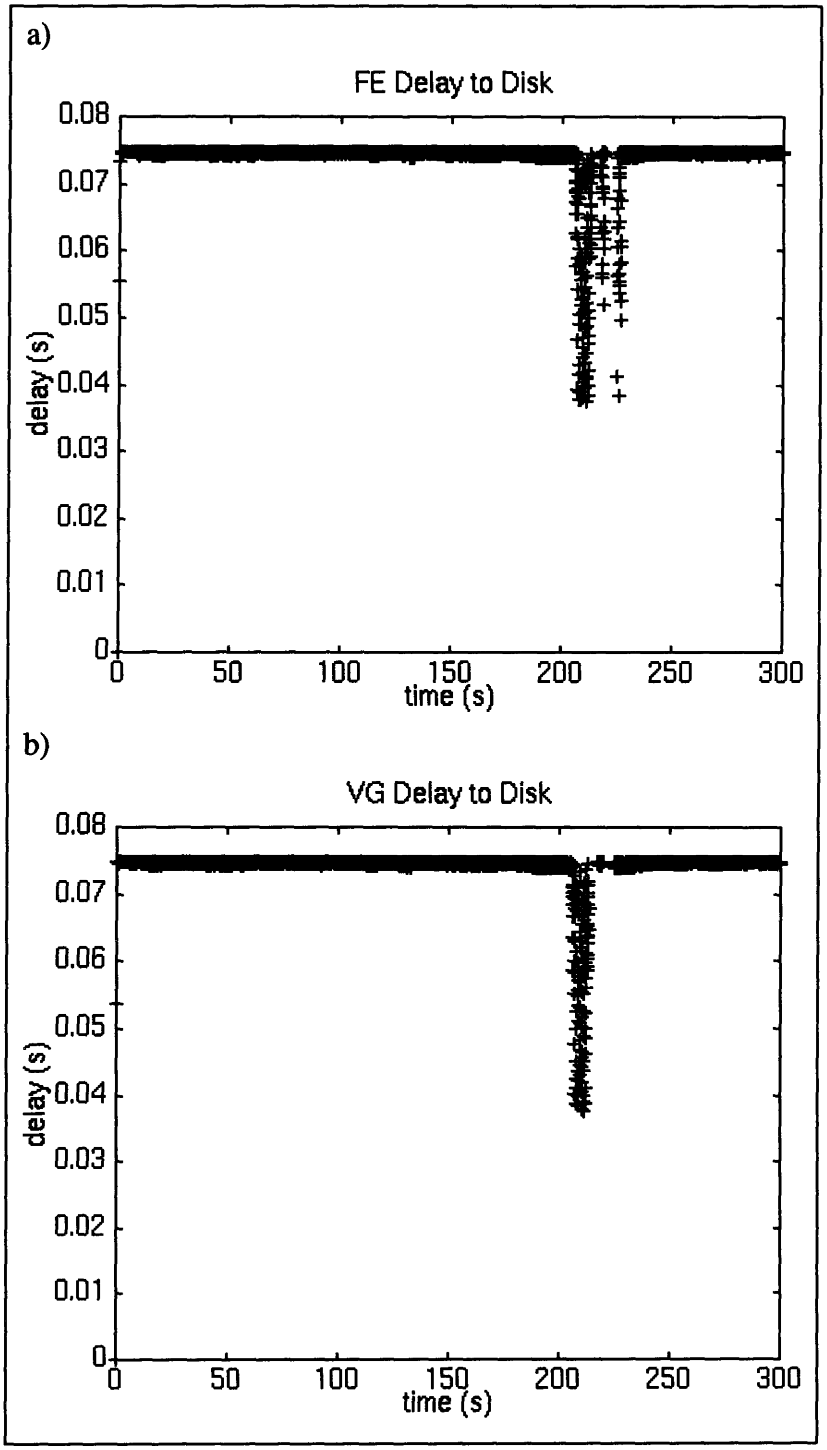

Figure 5.35: The delay to disk from the simulation involving 7 ultrasound machines, fast-wide SCSI-2, and a block size of 512KB.

The data received by the reviewstation over time is shown in Figure 5.36. As more

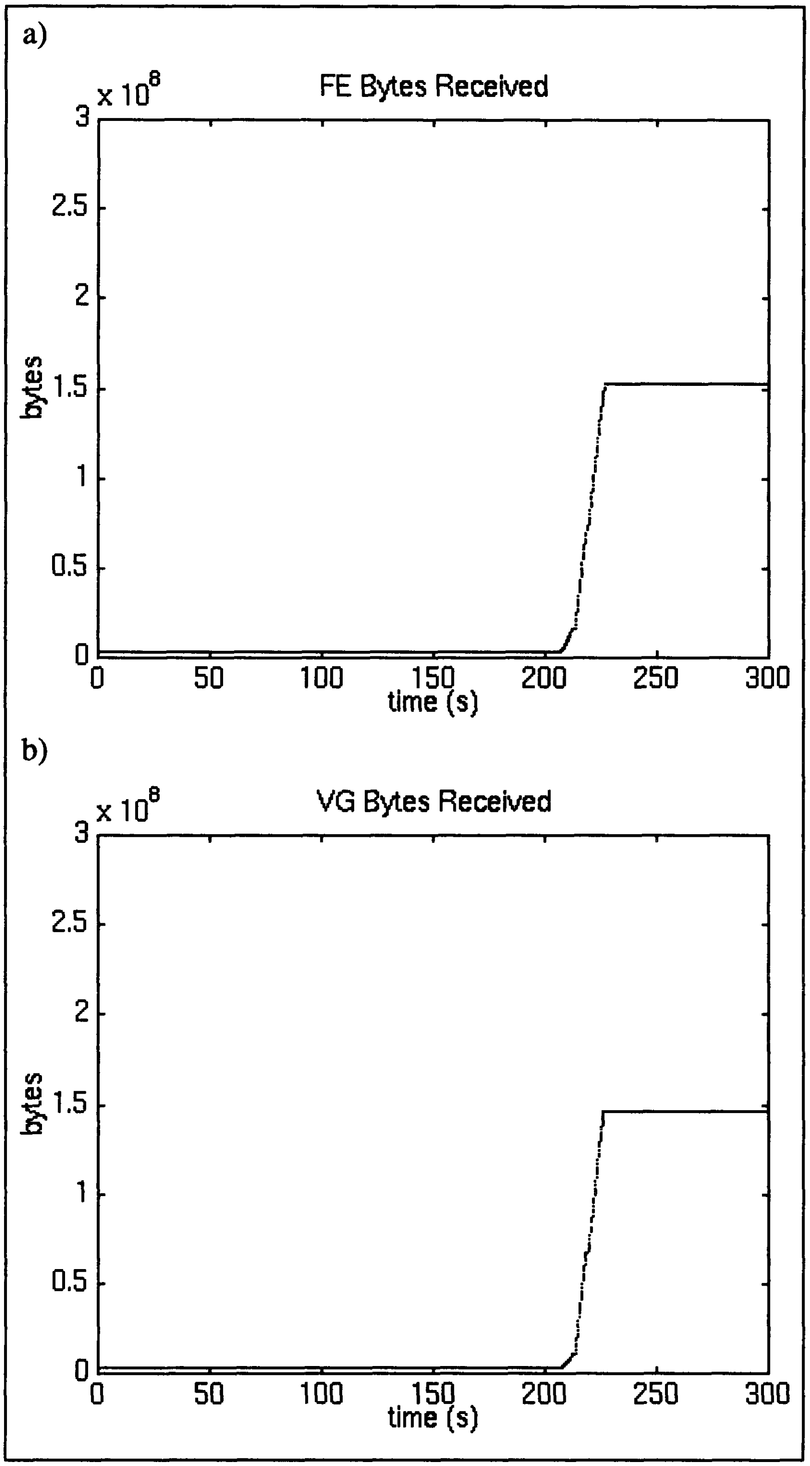

Figure 5.36: The data received by the reviewstation over time.

ultrasound machines are added to the network, the odds of having a time when the network is completely free of storage traffic is reduced. In this run, there was only one interval when none of the machines were generating data. As mentioned before, the fileserver model's policy relies on the existence of such total lulls to send data to the reviewstation, but they hardly ever happen. This argues for a change in the fileserver policy.

5.3 Summary

A number of simulation runs were conducted to see how the IMACS configuration choices of network medium (100VG or Fast Ethernet), speed of the SCSI-2 transfer from the fileserver to the disk (160 Mbps fast-wide SCSI-2, or 80 Mbps normal SCSI-2), and the block size of the transfer between fileserver and disk (4KB or 512KB).

The results can be summarized as follows:

- The maximum buffer sizes on the ultrasound machines will be significantly larger on Fast Ethernet than on 100VG (for example, the simulation with 5 ultrasound machines saw the Fast Ethernet ultrasound machine buffer hold ~30 packets, whereas the 100VG ultrasound machine held at most one).
- Because of backoff on the Fast Ethernet, the reviewstation on a Fast Ethernet may receive more data back from the fileserver than its counterpart on 100VG.
- The use of a small block size for the transfer from the fileserver to the disk will cause the fileserver buffer to grow immense, and prevent data from reaching the disk in a timely manner.
- Although a faster transfer rate helps information reach the disk sooner, the block size is a more important variable.
- The fileserver must follow a different policy of reading from and writing to the disk, than simply giving writes total preference.

Chapter 6

Conclusion

6.1 Project Results

In this thesis project, an investigation was undertaken into what can be expected when cardiology ultrasound machines are connected together on a small Image Management and Communication System over a shared Fast Ethernet or a switched 100VG network medium. Data on the real usage patterns of the ultrasound machines by sonographers was collected and analyzed. Discrete event models of the various devices on a small IMACS system were developed, including a model of the ultrasound machine which uses the empirically determined patterns of use to determine how it generates data over the course of a simulation.

6.1.1 Workload Characterization

The workload characterization proved useful in gaining an understand of the kind and amount of data which could be expected. The clinical ultrasound exams of adult and pediatric patients studied have shown that for a significant portion of the exam time, the ultrasound machine is idle. For the average adult exam studied, idle gap time made up 37.4% of the exam. For the average pediatric exam, 46.0% of the time was spent idling. The rest of the time is basically split between two dimensional imaging and color flow imaging. The doppler modalities (color flow and two dimensional) and the M-mode modality are hardly used at all. The significance of the prevalence of gap time in the exams is that the ultrasound machines are not continuously streaming data onto the network, but instead have a bursty pattern.

Overall, it is apparent that since the ultrasound machine's current usage pattern does involve a tremendous amount of gap time, even with multiple machines on a network, the instances when they all generate 2D data at the same time occur a small, but measurable, fraction of the time. Moreover, this is the *current* usage pattern derived from how sonographers use analog storage; when real digital systems are developed, sonographers will eventually store much less data. Thus, IMACS systems can be developed to handle the load from these machines.

6.1.2 Disk Block Size

As far as the fileserver is concerned, however, the network medium is of little consequence—a 100Mbps network is a 100Mbps network. In both networks, the fileserver receives all the data that the ultrasound machines are sending, even if the delay is potentially much greater on the Fast Ethernet. The bottleneck in the system is not the network, but the fileserver. The machines are generating a tremendous amount of data, and if the fileserver is using a small block size for the transfer to the disk, it can never expect to clear its buffer to the disk. A large block size must be used.

From the reviewstation's point of view, the network does not matter much either since the fileserver is the bottleneck with its policy of giving total preference to the writes to disk. It would be preferable to follow a policy of mixing writes and reads to allow both to be serviced better, though not necessarily fairly because preference should still be given to the writes to disk since they cause the fileserver buffer to grow greatly. It would also be better for the fileserver not to send short blocks to the disk, if it does not currently have enough for a block, but instead wait for more data to come in from the ultrasound machines. While the fileserver is waiting to fill these short blocks, it could service the reviewstation's request for exam data. Currently, it seems that the fileserver is only able to service the reviewstation when a total lull in the storage traffic occurs, that is when none of the ultrasound machines are generating data. As more machines are added to the system, the chances of having these lulls shrinks greatly. A better policy is needed to service the reviewstation.

6.1.3 Disk Transfer Rate

Obviously, having a faster bus will help get data from the fileserver to the disk faster. But the simulations show that the block size of the transfer to the disk is more of a limiting factor than the speed of the SCSI-2 bus, or the speed of the network medium.

6.1.4 Network Medium

A number of simulations of different configurations were run to determine how the ultrasound machines offer network load. The results of those simulation runs show that the network medium does affect the Sonos buffer significantly, since collisions on a Fast Ethernet involve a backoff, and can cause a large buildup of packets in the ultrasound machine buffer. The ultrasound machine buffers on the 100VG network, however, rarely suffer such buildups of packets. With five machines generating, the per machine difference between the maximum buffer size was about 40 packets; with 7 machines generating, the difference was about 400 packets.

Moreover, because of its Demand Priority policy, the 100VG suffers less interaction from the fileserver when it is sending data to the reviewstation. On the Fast Ethernet, the fileserver's servicing of the reviewstation may cause the ultrasound machines to backoff and buffer the data they were trying to send.

Although a 100VG network would be preferable, workable IMACS systems on Fast Ethernet can be developed. The only real trade-off appears to be that the ultrasound machines will require a higher local buffering capacity than they would on 100VG. The impact on the ultrasound machine buffers can be reduced, however, simply by staggering the imaging sessions as much as possible. Since the average exam time is ~15 minutes, one session can start on the hour, the next at 15 minutes past the hour, the third at 30 minutes past the hour. Staggering the imaging sessions would greatly reduce the network contention and the number of collisions.

Overall, the network is not the problem. The real system bottleneck is the fileserver, based on its ability to clear itself of the incoming data from the ultrasound machines, and its policy of how fairly it services the reviewstation's requests for exam data.

6.2 Future Work

In the course of working on this project, a number of ideas came up that would be interesting to follow up on in future work. Some involve modifying the simulation slightly and others involve changing some aspect of the approach taken in this thesis, since there are a number of areas for improvement with the simulation model itself:

- The model currently samples the measures of interest (various buffer sizes for example), but would generate more meaningful results if it were to also have the ability to keep track of local peaks in the real buffer size, as opposed to just reporting what it measures at the sampling times.
- Currently the model treats the digital storage as one huge disk, and writes to it as if it were a log-structured file system. A better simulation of the IMACS system would take into account that the ultrasound machines are all saving to different files because this would affect how the data is written to the disk.
- How multiple fileservers affect the system would also be an interesting issue to investigate through simulation, since the fileserver was found to be the system bottleneck.
- One interesting item to research that was not addressed in this thesis, is whether even larger block sizes for the transfer to the disk would be better or worse. Only two block sizes were investigated, but it would be interesting to see how, for example, the use

of a 1MB block affects the fileserver's buffer, and the amount of data received by the reviewstation.

- The policy used in the fileserver model greatly affected how the reviewstation received data over the course of the simulation. Different policies for intermixing writing and reading from the disk should be simulated in order to find a more efficient means of providing service to both the ultrasound machines and the reviewstation.
- It would also be interesting to see how other network mediums such as Fibre Channel affect the system's performance.

6.3 Summary

This thesis has shown that both 100VG-AnyLan and Fast Ethernet are workable solutions for the network medium of an Image Management and Communication System. Although 100VG-AnyLan has the advantage that the local buffering on the ultrasound machines can be much smaller, Fast Ethernet is a justifiable choice for an IMACS system. The system bottleneck is not the network medium, but the fileserver because of its policy for dealing with the reviewstation, and how it transfers data to and from the disk. With network mediums capable of 100Mbps throughput, the real problem for image management systems is not one of network bandwidth, but the system's ability to quickly write data to permanent storage.

Appendix A

Ultrasound Imaging Modalities

The model of the SONOS ultrasound machine presented in this paper views modes as: 2D, color flow, doppler, color flow doppler, freeze, and gap, because the model only cares about the amount of data coming out of the machine onto the network. The actual ultrasound modalities used by ultrasound machines, however, and those encountered in the studies of the clinical videotapes, are two-dimensional (2D) imaging, one-dimensional imaging (M-mode), Doppler imaging (Doppler), and color flow imaging (CF).

A.1 Two dimensional imaging (2D)

All ultrasound images are created from the returned echoes when pulses of ultrasound are sent into the body from a piezoelectric transducer. Two-dimensional (2D) imaging requires the use of a transducer containing an array of piezoelectric elements so that the transducer can send out multiple pulses at different points to create a scan line. A two-dimensional image (2D) is the cross-sectional image of tissue that appears when pixels corresponding to the returned echoes are illuminated to various degrees. The brightness (gray scale) is proportional to the echo strength. [Kremkau (3), 173]

2D imaging is the most heavily used modality. This technique allows for the visualization of the organ being imaged so that its shape and movement can be analyzed. Using two-dimensional imaging, for example, cardiologists can actually see whether valves in the heart are opening and closing properly, or see structural defects in the heart.

A.2 One dimensional imaging (M-mode)

M-mode, also called one-dimensional imaging or motion-mode, was once the mainstay of cardiology before the development of 2D imaging techniques. [Kung (4), 17] M-mode does not utilize an array of transducer elements like 2D imaging does; instead, m-mode uses a single piezoelectric element to repeatedly fire pulses of ultrasound. On the display, a pixel is brightened for each pulse as it returns to the transducer, showing a recording of motion versus time. [Kremkau (3), 183] M-mode is useful for analyzing the motion of a structure in a single direction, but offers no information on the motion in other directions at the same time, and offers no visual information of the shape.

A.3 Doppler Imaging

Of the two previous technologies discussed, 2D and M-mode, neither makes use of the Doppler shift of the received echoes, which can give substantial information to help in the examination of the heart. Doppler ultrasonography is useful in detecting and quantifying the presence, direction, speed, and character of blood flow in vessels. [Kremkau (3), 196] The Doppler flow information is displayed as either a flow versus time plot (Continuous Wave or Pulsed Wave Doppler) or as color-coded velocity information (Color Flow Doppler) in a two-dimensional image (see A.4 Color Flow Imaging.)

The Doppler effect is a change in the frequency or wavelength of a wave as a result of motion. The common example to illustrate the Doppler effect in everyday life is that of the police siren. As it approaches, the sound seems to get higher and higher in pitch; as it moves away from the listener, the pitch appears to reduce. The Doppler shift measured in medical ultrasound is simply the change in the frequency between the emitted pulses and the received echoes. [Kremkau (3), 196]

A.4 Color Flow Imaging

Color flow imaging is a Doppler technology that shows the motion of the blood flow, and can be used with two-dimensional imaging. Typically, the color flow information showing the flow or turbulence of the blood in color is overlaid on a grayscale two dimensional image of the cross section of the heart. Color flow imaging is very useful in finding abnormalities in blood flow such as areas of heavy turbulence or vortexes in the flow.

The blood flow through the volume being scanned is displayed in gradations of two colors, typically blue and red. One color is chosen to show the blood flowing towards the transducer, and the other is reserved to show the blood flowing away from the transducer. An example color scheme may set red to indicate flow towards the transducer, blue to indicate flow away from the transducer, and black to indicate unmoving blood. The level of the gradation of the color towards true red or true blue is used to indicate the velocity of the blood. For example, blood moving extremely fast away from the transducer would be colored bright blue, i.e in terms of the pixel's RGB values, red and green would contribute no amount, and blue the full amount.

Appendix B

Graphs of the Span Lengths

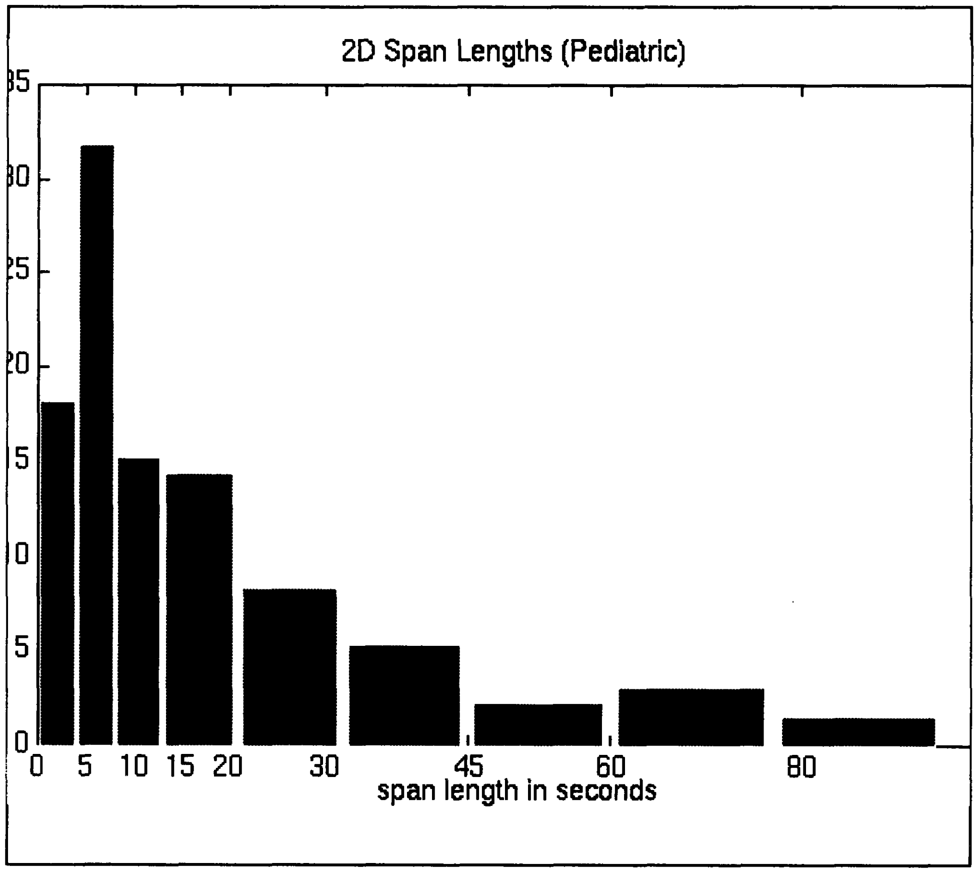

Figure B1: The 2D span lengths of the average pediatric exam.

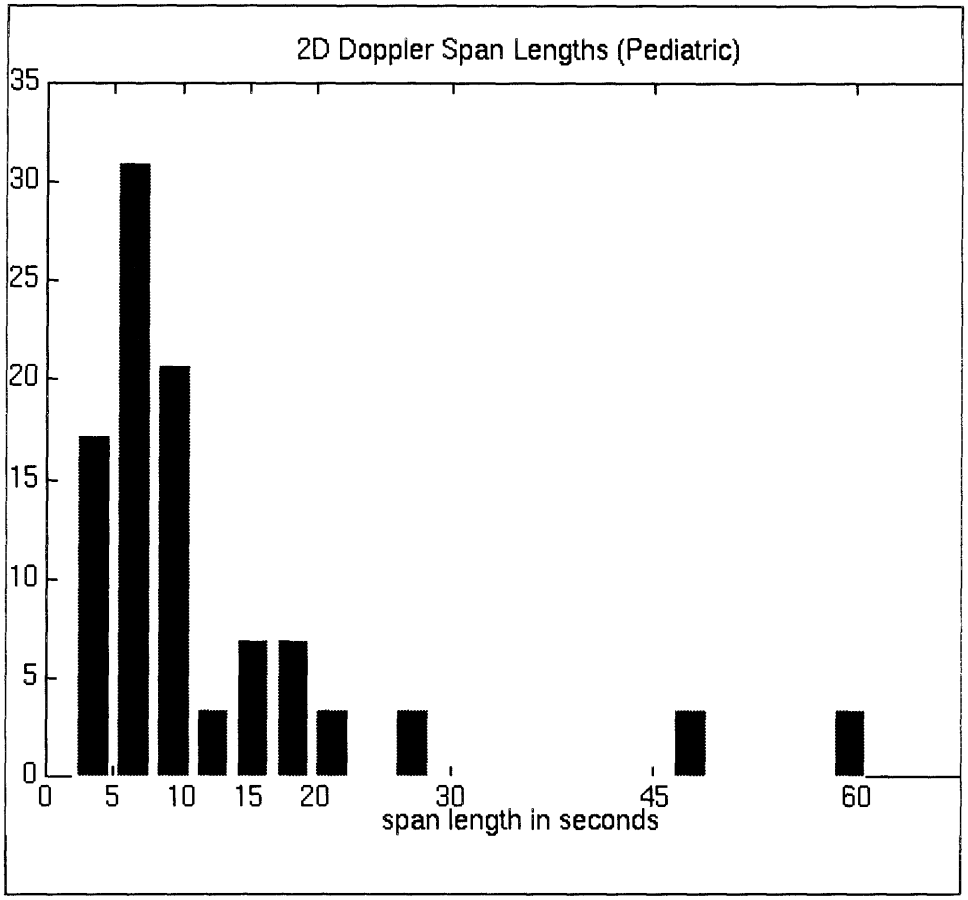

Figure B2: The 2D Doppler span lengths for the average pediatric exam.

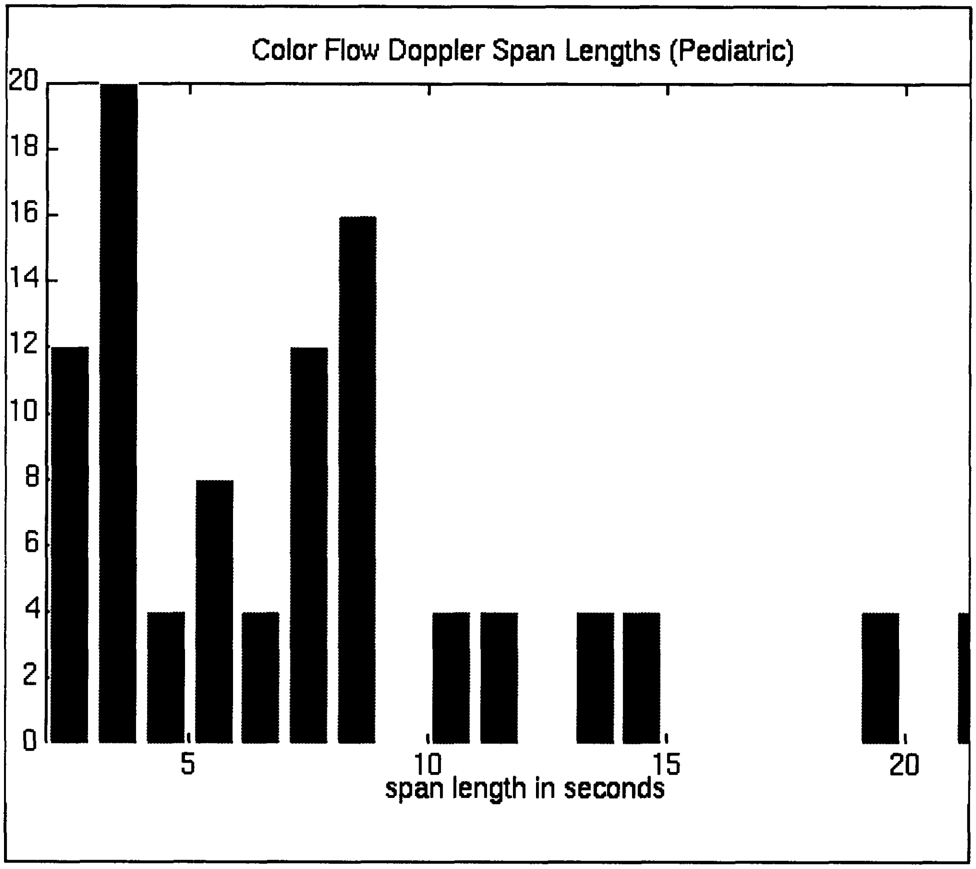

Figure B3: The CF Doppler span lengths for the average pediatric exam.

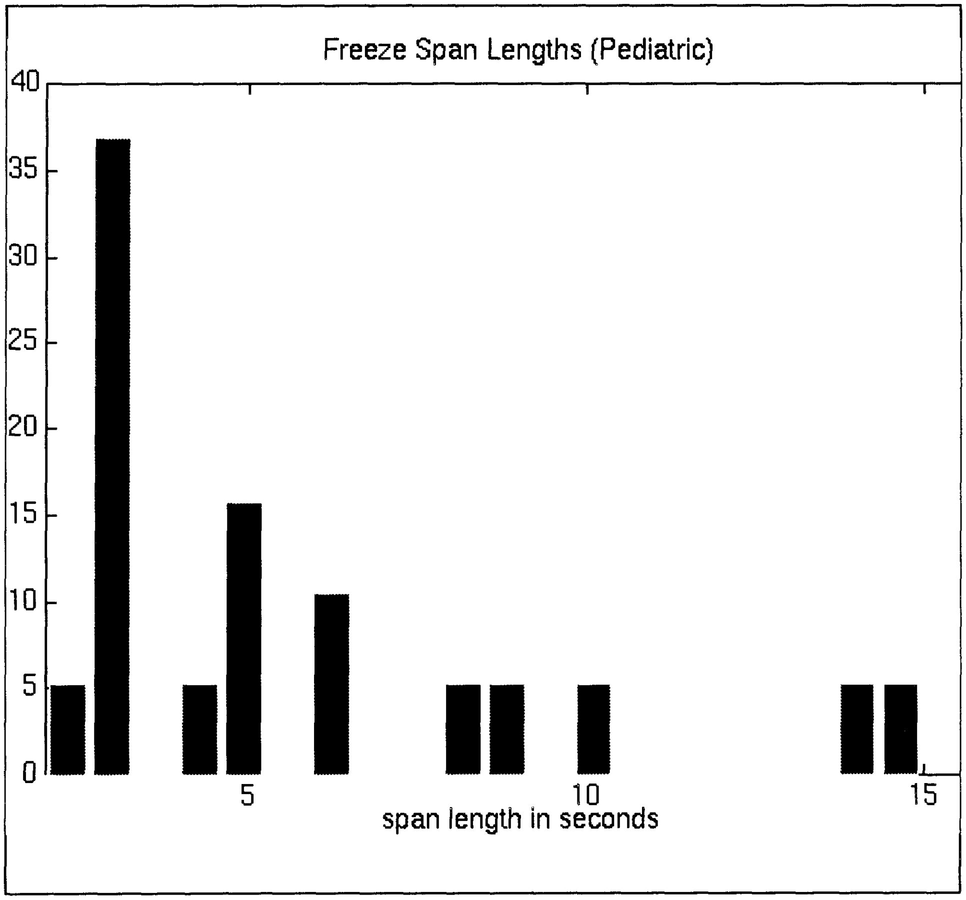

Figure B4: The freeze span lengths in the average pediatric exam.

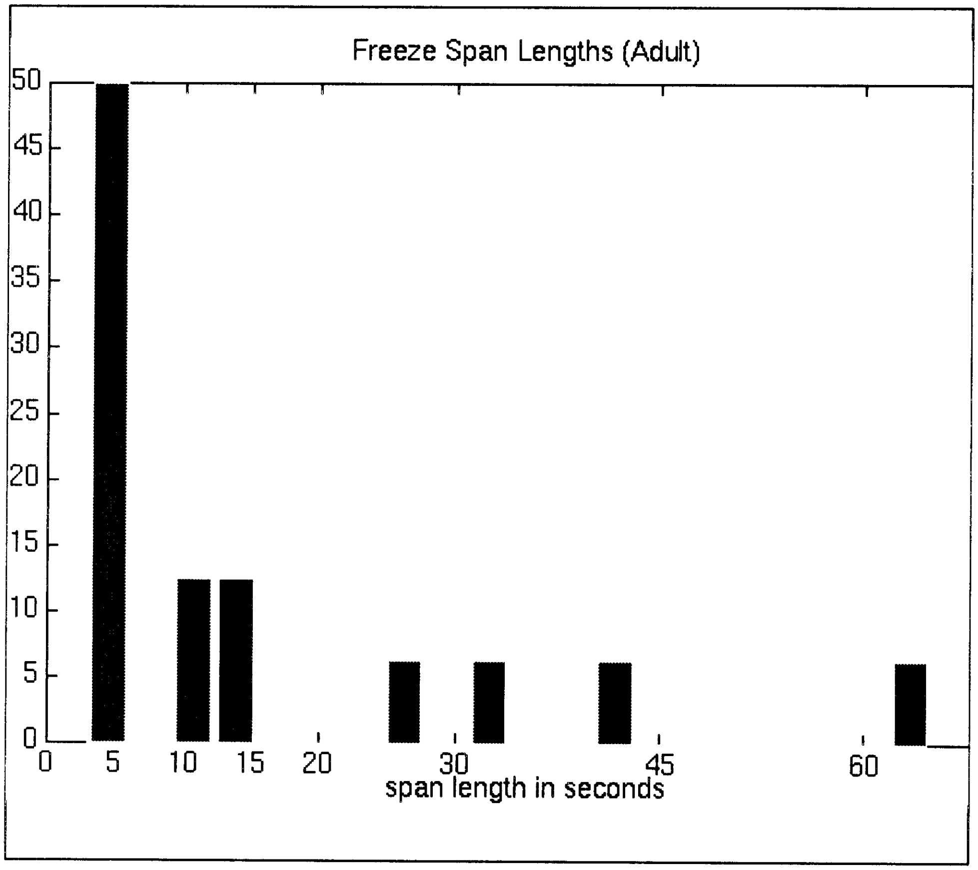

Figure B5: The freeze span lengths for the average adult exam.

References

1) Accredited Standards Committee XT39.2. *Small Computer System Interface-2 (SCSI-2).* (1990).

2) Anderson K., Meredith G., *et al.* "Hierarchical Rapid Modeling and Simulation of a High Performance Picture Archiving and Communication System (PACS)." *SPIE Medical Imaging VI: PACS Design and Evaluation.* (1992) 1654: 577-588.

3) Baraniuk S., Baxter N., *et al. OPNET Tutorial Manual.* MIL 3, Inc.(1993).

4) Davis W.A., Maydell U.M., and Gill S.G. "Simulation of a PACS in a hospital environment." SPIE Medicine XIV/PACS IV. (1986) 626: 740-747.

5) Delaney, D. "Improving Ethernet LAN Performance with Ethernet Switching." Plaintree Systems White Paper. (1995)

6) Farallon Computing, Inc. "Fast Ethernet White Paper: Introduction to 100Base-T." (1995).

7) Feigenbaum H. "Digital Recording, Display, and Storage of Echocardiograms." *Journal of the American Society of Echocardiography.* (Sep/Oct 1988) 1(#5): 378-383.

8) Garzia R., Garzia M. *Network Modeling, Simulation, and Analysis.* Marcel Dekker, Inc. New York. (1990)

9) Hewlett-Packard Company. "100 VG-AnyLan: A Technical Overview." Hewlett-Packard Application Note. (1994)

10) Hewlett-Packard Company. "HP C2244/45/46/47 3.5" SCSI-2 Disk Drive Technical Reference." (September 1992)

11) Huang H.K., Kangarloo H., *et al.* "Planning a Totally Digital Radiology Department." *AJR.* (Mar 1990) 154: 635-639.

12) Jain R. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling.* John Wiley & Sons, Inc. (1991).

13) Kung D. *Prototype Software Environment for Digital Ultrasound Review.* S.M. Dissertation. Massachusetts Institute of Technology. (1994).

14) Lawrence G.R., Marin G.A., Naron S.E. "Hospital PACS (Picture Archiving and Control System) Network Simulation Studies" *SPIE Medicine XIV/PACS IV.* (1986) 626: 729-739.

15) Martinez R., Sanders W.H., *et al.* "Performance evaluation of a Picture Archiving and Communication System using Stochastic Activity Networks." *SPIE Medical Imaging VI: PACS*

Design and Evaluation. (1990) 1234: 167-178.

16) Pavicic M., Ding Y. "Multicomputer performance evaluation tool and its application to the Mayo/IBM image archival system." *SPIE Medical Imaging V: PACS Design and Evaluation.*(1991) 1446: 370-377.

17) Ratib O. "Geneva PACS Project." Digital Imaging Unit. University Hospital of Geneva, Switzerland.

18) Saulnier E.T., Barnett B.G. "High-level Analysis of Medical Imaging Traffic." *SPIE PACS Design and Evaluation.* (1993) 1899: 260-270.

19) Saulnier E.T., Bortscheller B.J. "Impact of Workload Model on PACS Performance Prediction." *SPIE PACS Design and Evaluation.* (1993) 1899: 178-185.

20) Stallings W. *The Business Guide to Local Area Networks.* Howard W. Sams & Co, 1990.

21) Stewart B.K., Lou S.L., *et al.* "An Ultrafast Network for Communication of Radiologic Images." *AJR.* (Apr 1991) 156: 835-839.

22) Stewart B.K., Lou S.L., *et al.* "Performance Characteristics of an Ultrafast Network for PACS." *SPIE Medical Imaging V: PACS Design and Evaluation.* (1991) 1446: 141-153.

23) Stewart B.K., Dwyer S.J. "Teleradiology System Analysis Using a Discrete Event Driven Block Oriented Network Simulator." *SPIE Medical Imaging VI: PACS Design and Evaluation.* (1992) 1654: 2-13.

24) Stut W.J.J, de Valk J.P.J., *et al.* "Modelling and Simulation within the Dutch PACS Project." *SPIE Medical Imaging III: PACS System Design and Evaluation.* (1989) 1093: 423-428.

25) Stut W.J.J., van Steen M.R., *et al.* "Simulation-based PACS development." *SPIE Medical Imaging V: PACS Design and Evaluation.* (1991) 1446: 396-404.

26) Thomas-Conrad Corporation. "High Speed Networking." Thomas-Conrad Corporation White Paper. (1995)

27) Toshimitsu A., Fukushima Y., *et al.* "Network Data Rate Requirement Analysis for Picture Archiving and Communication Systems." *SPIE Medical Imaging IV: PACS System Design and Evaluation.* (1990) 1234: 147-158.

28) Van Meter, R. The comp.arch.storage FAQ, part 2. (1995)

29) Wolfman N.T., Boehme J.M., *et al.* "Evaluation of PACS in Ultrasonography." *J Ultrasound Med.* (1992) 11:217-223.

30) Wong A.W.K., Stewart B.K., *et al.* "Multiple Communication Networks for a Radiological

PACS." *SPIE Medical Imaging V: PACS Design and Evaluation*. (1991) 1446: 73-80.

31) Wong A.W.K., Huang H.K. "Subsystem Throughputs of a Clinical Picture Archiving and Communication System." *Journal of Digital Imaging*. (Nov 1992) 5(#4): 252-261.

32) Wong A.W.K, Ricky K.T, Huang H.K. "Digital Archive Center: Implementation of a Radiology Department." *AJR*. (Nov 1992) 159:1101-1105.